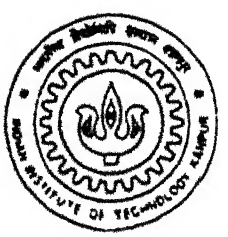
DESIGN OF LOW BITRATE VIDEO CODER USING 3D MOTION ESTIMATION

by

Karthik M



DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

March, 1999

Design of a Low Bitrate Video Coder Using 3D Motion Estimation

A Thesis Submitted

in Partial Fulfilment of the Requirements

for the Degree of

Master of Technology

by

KARTHIK M

to the

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

March, 1999

Certificate

It is certified that the work contained in the thesis entitled DESIGN OF A LOW BITRATE VIDEO CODER USING 3D MOTION ESTIMATION, by Karthik M, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

(Dr. Sumana Gupta)
Associate Professor
Department of Electrical Engineering
Indian Institute of Technology, Kanpur

26^{th}March, 1999.

To

My Brother

Acknowledgements

I would like to express my sincere gratitude to Dr. Sumana Gupta for her invaluable guidance and constant encouragement in completing my thesis work.She has been very helpful throughout the duration of the thesis.

I wish to thank Silicon Automation Systems for their financial support in the fourth semester.A special word of thanks to Narsi for all his image processing routines which forms the basic block of my programs.

I wish to thank my Kannada friends, Craja and RC Swamy who have made my stay at IIT Kanpur a good experience I would love to mention the names of my friends Ashok, Sganguly, Antu, Srdas, Rkpila, Jay, Pratul,Siddha, Mama , who have made my stay at IITK, an unforgettable experience. But for these people ,this thesis would have finished in half the time .

I specially wish to thank Sandhitsu(sandy) who saw to it that I didn't learn much about computers by solving all my computer-related problems. It is only appropriate that I mention Shafi who was the final resort for all computer-related problems which were beyond the scope of anybody else.

Last but not the least I wish to acknowledge the constant support, blessings and guidance which my family gave me.

Abstract

In this thesis we describe the design of a low bit rate video coder using a 3-D motion estimation algorithm based on the E-matrix method. The error between the current frame and the motion compensated previous frame was transformed using the shape adaptive DCT(SADCT). This method is specially suitable for coding boundary regions in an image. The transform coefficients using SADCT is considerably less compared to DCT for such regions. The transform coefficients were quantized into levels varying from -127 to +127 which were subsequently VLC coded. The motion parameters were FLC coded. Both codes were sent as serial bits to the receiving side where they are decoded.

Reasonably good quality reconstructed image sequences at bit rates of the order of 63kbps, 61kbps and 57kbps were obtained by suitably changing the dead zone and step size of the quantizer. The coder was tested using the standard **Claire, Miss America and Salesman** sequences. The PSNR obtained in each case ranges from 37 to 31 indicating good quality of reconstructed images. The model failure areas varies from 4% to 7%.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The need for video compression

The digital representation of an image or an image sequence requires a large number of bits. To illustrate this we take the example of displaying a good quality video on a personal computer. The standard size of the image for this is 640×480 pixels. Each pixel requires 3 bytes for representing the three primary colors at that pixel. So a single image requires $640 \times 480 \times 3 \approx 900$Kbytes of data. In order to get the video effect, the images are played in succession at a certain rate. Normally this rate is 30 images/sec. This implies that we need to store $900 \times 30 = 27$ Megabytes for every second of video on disk. In addition to this storage requirement, 27Mbytes/sec of disk space is needed for the storage systems (retrieval rate) and the display monitor (delivery rate) The problem of video delivery worsens if one expects video to be delivered from a network rather than the disk. This is because network bandwidths are lower by several orders of magnitude than the local bus system on which a disk and a display

system reside.

To solve the twin problem of storage and delivery of video, digital still image compression and digital video compression techniques have emerged. The main goal of image/video compression is to reduce the required bit rate as much as possible and to reconstruct a faithful duplicate of the original image. Digital image compression deals with the problem of compressing a single image by taking advantages of the redundancies present in it - such as areas of similar color.

The efficient digital representation of image and video signals has been the subject of considerable research over the past 20 years. Digital video coding technology has developed into a mature field and products have been developed that are targeted for a wide range of emerging applications, such as video on demand, digital TV/HDTV broadcasting and multimedia image/video data base services. The increase in commercial interest in video communication gives rise to the need for international image and video-compression standards

.

To meet this need, the moving Picture Experts Group(MPEG) was formed to develop coding standards[12]. MPEG-1 and MPEG-2 video coding standards have attracted world-wide attention . An increasing number of very large scale integration (VLSI) and software implementations of these standards have become commercially available. MPEG-4, the most recent MPEG standard for transmitting video signals at very low bit rates is motivated by its potential applications for video phones, video conferencing, multimedia electronic mail, remote sensing, electronic newspapers, interactive multimedia databases, multimedia annotation, surveillance, telemedicine, communication

aids for deaf people and many others.

The main hurdle in the implementation of these applications arises from the difficulty in compressing huge amount of visual information to meet the available bandwidth range. In video phone or video conferencing applications the background of the scene remains unchanged in consecutive frames and only few moving regions are present. So there is a lot of redundant information which is removable. As a result a video sequence can be represented by a low bit rate stream which can be transmitted over a telephone network whose bandwidth is of the order 64Kb/sec. Let us first discuss a few common video standards

1.2 Existing Video Standards

1. *ITU-T H.261 Video Coder*

   The ITU-T H.261 Video Coding Standard came into existence in early 80's.This video coder primarily aims at achieving bit rates of $p \times 64$kbps where p varies from 1 to 30.Motion compensation is employed to predict the images.DCT and VLC are also used .The main applications are Video-telephony and Video Conferencing

2. *MPEG-1*

   MPEG1 was the next in line after H.261.Many storage media and telecommunication channels like LANS etc are well suited for a bit rate of 1.5 Mbps.MPEG1 satisfies these requirements.In addition,it is also the first standard to jointly implement both the video and audio coders.It encourages a lot of interactive applications.It doesn't specify any stan-

dard encoding process.Hence the compression algorithm is left to the designer.But the image quality should be comparable to that of a VCR and the audio quality to that of a CD.

3. *MPEG-2*

This is basically a a full motion audio and video coding standard which meets a number of requirements.It has a variable bit rate ranging upto 100Mbps.It supports a lot of interactive applications like Digital storage media,Computer graphics,multimedia,video games EDTV HDTV etc.Hence it provides a generic solution to video/audio coding (transmission and storage).This generic coding standard is designed to provide a wide spectrum of bit rates ,resolutions,quality levels and services. The advantage of this system is it's compatibility and scalability with other systems.The four compatibilities are defined as :

- A system is called as backward compatible if an existing decoder can decode a signal encoded by the new encoder
- A system is forward compatible if a new decoder can decode a signal encoded by the present encoder
- The system is upward compatible if a higher resolution decoder is able to decode the signal encoded by the present system
- downward compatibility is described similarly

MPEG-1/H.261 are forward and upward compatible with MPEG-2.The standard MPEG-2 is backward and downward compatible with MPEG-1/H.261.As far as audio is concerned MPEG-2 is only forward and backward compatible. Scalability means that a part of a bit stream can be decoded.This allows decoders with less processing power to display video at lower resolution/quality.The video standard MPEG2 is scalable.

4. *MPEG-4*

   MPEG-4 was started with a view of achieving bit rates lower than 64kbps to enable the transmission of video and audio through Public Switch Telephone Networks.This is how MPEG-4 evolved beginning from H.261.The final draft is yet to be released.

1.3 Image Compression Methods for Video Phone Image Sequences

The video telephone problem can be defined as the problem of compressing the huge amount of visual information into a very low bit rate stream for transmission over public switched telephone network (PSTN). It is known that the available PSTN network is mainly used for transmission of speech. Visual data is considerably larger than the speech data. If we adopt the CIF (common intermediate format) standard, for which the spatio-temporal resolution is $360 \times 288 \times 30$, then the bit rate required is approximately 74 Mb/sec. On trying to transmit a color video signal via the PSTN under the assumption that channel capacity is extended to 64 Kb/sec and using 56 Kb/sec for video and another 8 Kb/sec for voice, the compression ratio needed is higher than 1000. Achieving such a high compression ratio indeed poses a serious challenge to the researchers in the field of image coding .

It is not possible for TV video signal to attain such high compression ratios while maintaining a reasonable quality of the decoded images for transmitting over PSTN. But for video telephone applications such high compression ratios can be attained by exploiting some special features of the scenes. Typical video

phone scenes have the following three special characteristics:

1. **Fixed scene content** The typical scene is a head and shoulder image of the speaker. Due to the objects in the scene being known *a priori*, some knowledge about them can be used.

2. **Limited motion** The interfering motion mainly caused by the movement of the speaker and the camera is generally fixed. This situation is different for mobile video telephone, as in this case the camera undergoes limited motion, such as zoom, pan and vibration. The movement of the speaker mainly consists of the global movement of the head and shoulder and the local motion due to changes in facial expression. Due to the inertia of the human body, the global motion is relatively slow and can be described using only few bits per frame. In this way, more bits can be spent on facial expressions .

   In our case we transmit only the head and shoulder region of Claire as the background scene content is fixed.

3. **Special requirements of visual information** Interpersonal video communications does not usually require the full resolution that is provided by broadcast television. The key to visual communication is to provide the emotional dimensions. Therefore, a lower resolution image format is often used.

One commonly used format is the QCIF in which the resolution is reduced to 144 × 180 for luminance and 72 × 90 for chrominance. The frame rate is reduced to 10Hz or even 6Hz. The combined content of the knowledge of the scene, the spatio-temporal redundancy etc. allows the visual information to

be compressed to obtain a very high compression ratio. A brief review of the techniques for very low bit rate image sequence coding is discussed in this section.

R.Wallies proposed an ultra-low bandwidth video conferencing system which could be operated at 9.6Kb/s bit rate. The compression is achieved by transforming the original grey level image into a binary image. 2-D run-length coding techniques are then used to compress the binary images. Lippman [1] proposed two approaches to transmit a video phone scene. In the first approach referred to as storage based coding, the images to be displayed are known in advance and transmitted with full resolution and full dynamics by local retrieval. In the second scheme only the instructions. to retrieve the images or sequences are transmitted. These coding schemes are aimed at 8Kb/s. and have low resolution with a low frame rate.

For video conferencing, these drawbacks are overcome by allowing a transmission bit rate of 64Kb/s. Methods used to achieve this can be divided into two categories:

- waveform-based coding and
- model-based coding

1.3.1 Waveform-based Coding

In waveform-based coding image and video compression is achieved by exploiting the inherent statistical redundancy of the image data. Most image coding techniques such as transform coding, subband/wavelet coding, VQ coding and

fractal coding can be included in this group. These methods cause coding artifacts known as blocking and mosquito effects.

1.3.2 Model-based Coding

In model-based image coding the input image is viewed as a 2-D projection of a 3-D real world (scene). The coding is performed by first modeling the 3-D scene, extracting the model parameters at the encoder and finally synthesizing the image at the decoder by using the extracted and quantized parameters. If we can reconstruct the three dimensional scene model that leads to 2-D image sequence, and the images are analyzed and synthesized based on this model, then a great reduction in image information can be expected. This is the basic idea of the *model-based coding* method . However they can be used only for a head and shoulder scene or for a very limited range of scenes.These coders offer the promise of fairly good quality images at low bit rates,but the complete systems are still under study and the coders designed require considerable computational resources.Need for a good model is very important since the quality of image reconstruction depends,to a large extent,on the model.Furthermore,the reported systems are limited to a particular model,such as head and shoulders and a stationary background.A system using a model of **Claire** can not provide good results for the **Miss America** sequence.But the bit rate can be reduced while preserving the quality.

In these methods we employ a concept called motion estimation and compensation and it is the error that is transform coded(this part is quite similar to transform coding).

In this thesis we develop a new 3D motion estimation algorithm which estimates the rotational ,translational and 3-D spatial positions of a set of points whose point correspondences are given.Though this method is a model based approach ,it is yet to be given the full functionalities of a model based coder.

The advantage of this method is that it can model any 3-D scene in general.

1.4 Organization of the Thesis

This thesis is organized as follows.

1. **Chapter 2** : This chapter discusses the new motion estimation algorithm and the associated motion compensation.Each image is estimated from it's previous image and the difference between the present image and it's estimated one is fed as input to the coder.

   The 3D motion parameters are obtained using the **A-Matrix and E-Matrix** methods respectively.Each of these matrix methods are explained in detail.

2. **Chapter 3** : This chapter describes in detail the SADCT based encoder and decoder.The error from motion estimation is encoded and transmitted through the receiving channel.The decoded images of different image sequences at different bitrates are displayed .

3. **Chapter 4** : This chapter shows the results obtained and compares the PSNR of different image sequences at different bitrates.

4. Chapter 5 : Chapter 5 concludes the thesis and discusses the scope for future work.

Chapter 2

Motion Estimation

2.1 Introduction

This chapter discusses a new 3D motion estimation algorithm using the E Matrix method to motion compensate the images.We review the need for motion estimation and the different types of 2D and 3D motion estimation algorithms.It is a common practice to estimate an image frame and code only the error of estimation.

The concept of motion compensation is based on the estimation of motion between video frames ie..,if a group of pels move in a particular direction their motion can be described by a motion vector.However it is unnecessary to estimate the motion vectors of all pels, as it is assumed that in a group of close pels, all pels have the same motion .So only the motion of the centre of a $N \times N$ block is estimated ,coded and transmitted.At the receiving side the entire block is assumed to have the same motion as it's centre and displaced

accordingly when motion compensation is employed.

Figure 2.1 shows the block diagram of a video coder [12] .Each image is reconstructed on the transmitting side by the Motion Compensator from the previous image .In other words,the motion Compensated $(N-1)^{th}$ image is the N^{th} reconstructed image on the transmitting side.The N^{th} incoming image and the N^{th} reconstructed image are subtracted to give an error image. This error is transmitted by VLC along with the motion parameters The error is recovered on the receiving side and the $(N-1)^{th}$ image (present in the Frame Store on the receiving side)is motion compensated by the received motion parameters to produce an image which is quite similar to the N^{th} reconstructed image on the transmitting side. The recovered error is added to this to give a faithful representation of the N^{th} original input image.

Hybrid DCT/DPCM Coding Scheme

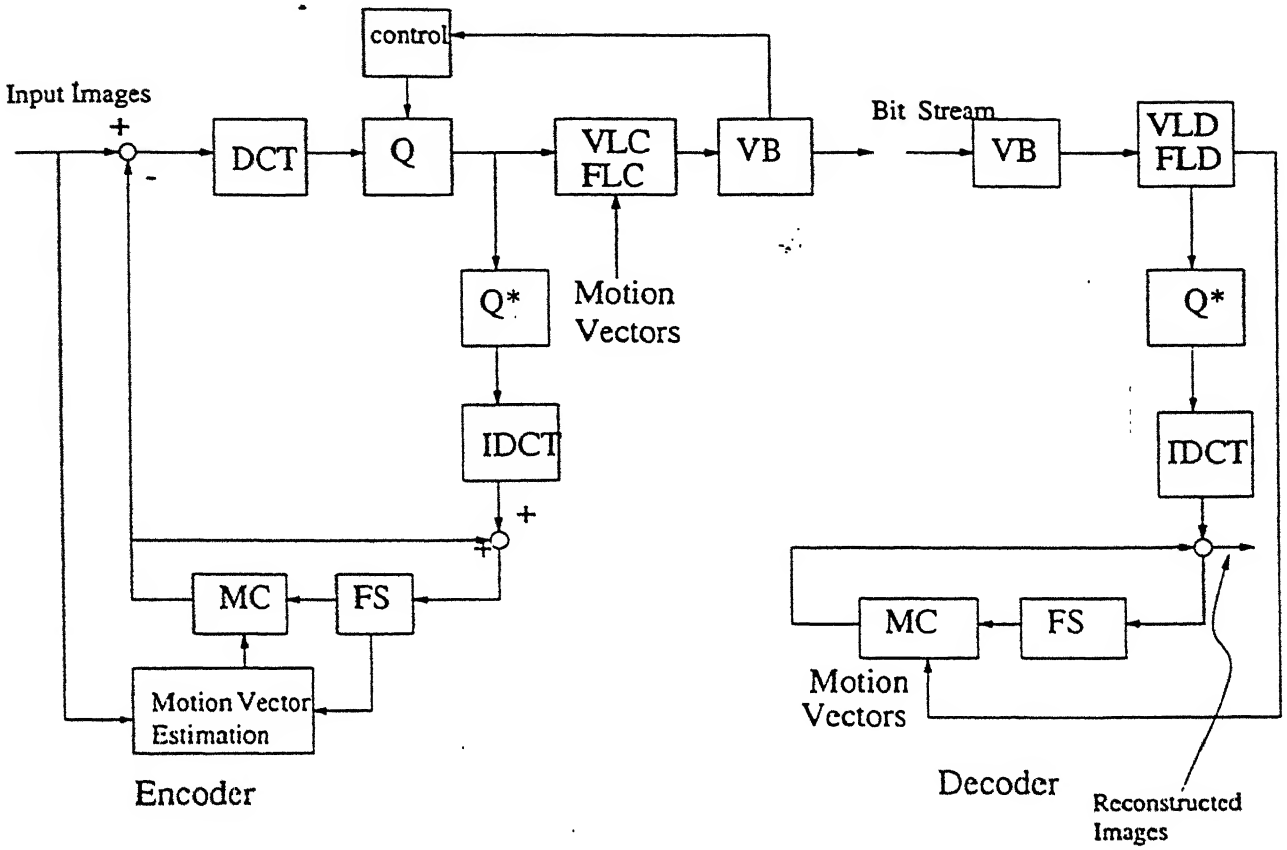


Figure 2.1: Block Diagram of the Video Coder

where

DCT:Discrete Cosine Transform Q;Quantizer Q*:Dequantizer

IDCT:Inverse DCT FC:Frame Store MC:Motion Compensator

VLC:Variable Length Coding VB:Video Buffer VLD:Variable Length Decoder

FLC:Fixed Length Coding(for motion parameters) FLD:Fixed Length Decoding

2.2 Motion Estimation

The standard 2D method used for Motion Compensation(MC)is the Block Matching method .

2.2.1 2D Motion Estimation

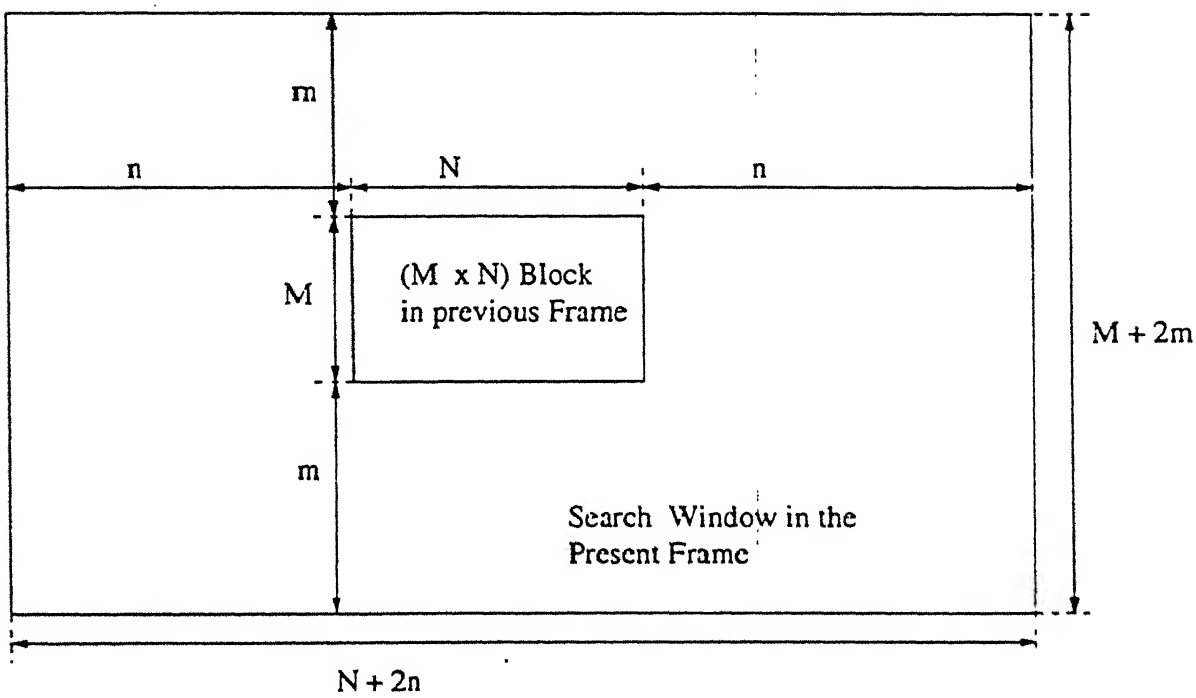


Figure 2.2: Motion Estimation of a $M \times N$ block in frame (N - 1) within a $[(M+2m) \times (N+2n)]$ search area in Frame N

We refer to figure 2.2.The motion parameters estimated here are the 2D vectors of the centre of the $(M \times N)$ Block in frame (N - 1) .This is done by calculating that value of i, j which minimises $M(i,j)$ defined as,

$$M(i,j) = \sum\sum_{(a,b)\in v}(X_{a,b} - X^{R}_{a+i,b+j})^2 \qquad |i| \leq m, |j| \leq n$$

where $(a, b) \in \upsilon$,the region of the $(M \times N)$ block in frame (N - 1),$X_{a,b}$ denotes the value of pixel (a, b) in frame (N - 1) and $X^R_{a+i,b+j}$ denotes the value of pixel $(a + i, b + j)$ in frame N. The position of pixel (a,b) when it is displaced by i units vertically and j units horizontally is $(a + i, b + j)$.Hence the summation can be viewed as the sum of errors between the $(M \times N)$ block in frame (N - 1) and a similar block in Frame N,whose centre is displaced by (i, j) from the centre of the block in frame (N - 1).

The value of (i, j) for which the above summation is minimum is the motion parameter (2D displacement vector) of the centre of the $(M \times N)$ block in frame (N - 1).Other measures like Mean Absolute Error,Cross-Correlation etc are often used instead of the Mean Squared Error criterion.A comparison of the performancers of the various measures is described in [9, pages 112].

2.2.2 3D Motion Estimation

Two types of projections are defined. They are

- Perspective Projection : In this this method of projection a 3D point with coordinates (x, y, z) as shown in figure(2.3) gets projected onto the point $(\frac{x}{z}F, \frac{y}{z}F)$ in the image plane. The point of intersection of the line joining the 3D point with the origin and the image plane gives the position of the projected point on the image plane.The image lies on the image plane which is perpendicular to the z axis and it's equation is $z = F$

  In this case the equations that relate the motion and structure parameters to the image plane coordinates are non-linear in the motion parameters.Early methods to estimate the motion and structure param-

eters required an iterative search which often lead to either divergence or convergence to a local minima.We use two step linear algorithms.With eight or more point correspondences we estimate some unknowns called as the essential parameters .The actual motion and structure can be derived from these parameters. We first present the **Epipolar Constraint** and define **Essential** parameters in section 2.2.3 This projection is followed when the variation in depths is comparable to the average depth.

- Orthographic Projection : Here a 3D point with coordinates (x, y, z) will be projected onto a point on the image plane with coordinates (x, y).A perpendicular is dropped from the 3D point onto the image plane and the point of intersection is the position of the projected point on the image plane.

2.2.3 E Matrix Method

Some notations and Equations

Perspective projection is assumed ant Focal length F is normalized to 1. Let $\mathrm{x} = (x, y, z)^T$ denote the 3D coordinates of a point before motion(time t1)

$\mathrm{x}' = (x', y', z')^T$ denote the 3D coordinates of the same point after motion(time t2)

$\mathrm{X} = (u, v, 1)^T = (\frac{x}{z}, \frac{y}{z}, 1)^T$ denote a vector of image plane coordinates at time t1

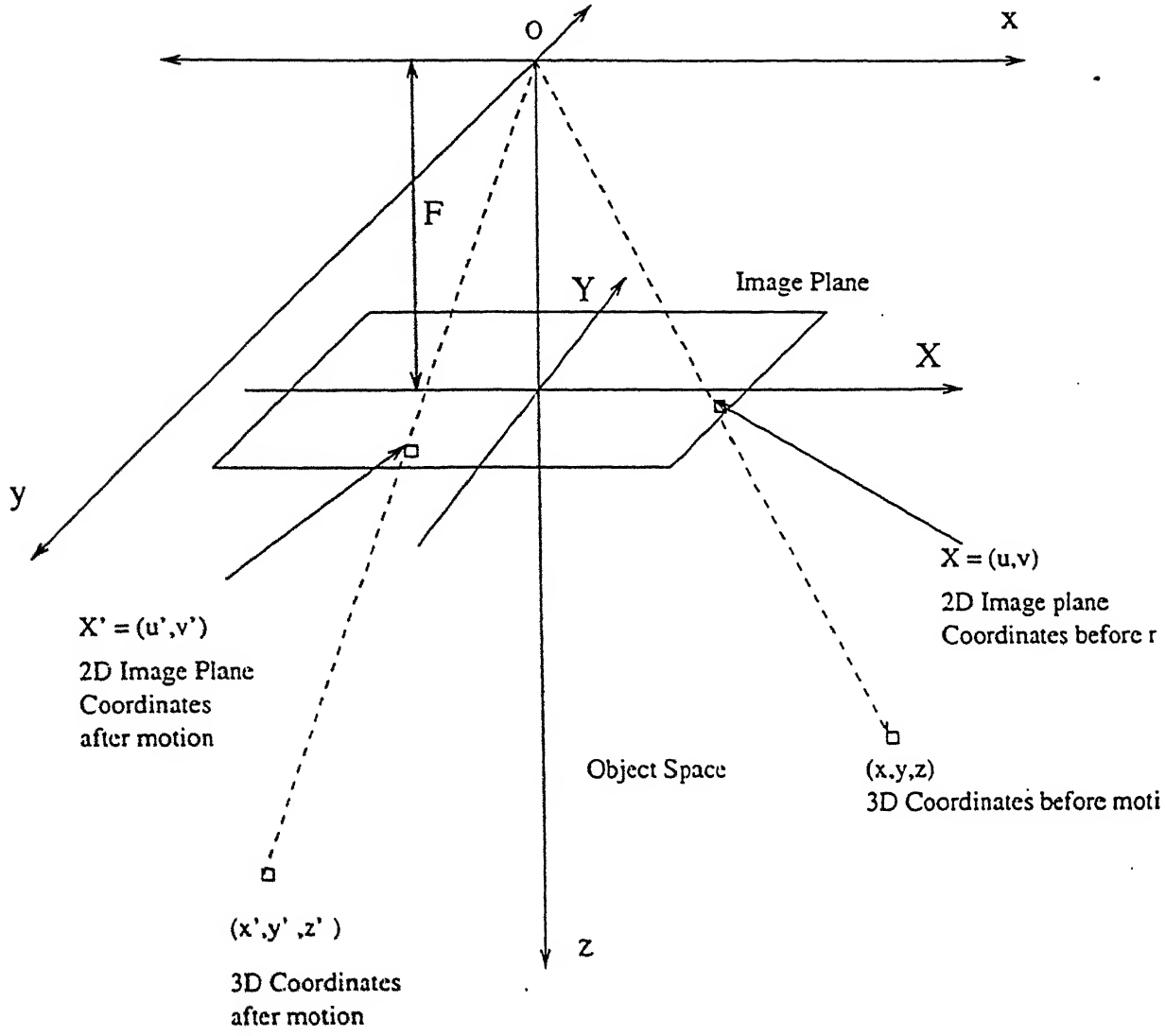


Figure 2.3: The basic perspective model

$\mathbf{X}' = (u', v', 1)^T = (\frac{x'}{z'}, \frac{y'}{z'}, 1)^T$ denote a vector of image plane coordinates at time t2

The relationships between the above vectors and motion,according to Per-

spective laws are:

$$\mathbf{x} = z\mathbf{X} \tag{2.1}$$

$$\mathbf{x}' = z'\mathbf{X}' \tag{2.2}$$

$$\mathbf{x}' = R\mathbf{x} + T \tag{2.3}$$

where R and T are the rotational and translational matrices of dimension 3 x 3 and 3 x 1 respectively

$$R = [r_1 \;\; r_2 \;\; r_3] = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{2.4}$$

$$T = \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} \tag{2.5}$$

We define norm of a matrix A

$$\|A\| = \sqrt{\sum_{ij} a_{ij}^2} \tag{2.6}$$

Also we define a 1-1 mapping of a 3 x 1 vector to a 3 x 3 matrix to facilitate easy calculation of cross-product of two vectors as,

$$[(a\ b\ c)]_X^T = \begin{bmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{bmatrix} \tag{2.7}$$

Now Cross product of any 2 vectors X,Y is defined as the following Matrix Multiplication

$A\ X\ B = [A]_X B$ where $[.]_X$ is the above mapping

Also we define

$$\hat{T} = \frac{T}{\|T\|}$$

as the unit translational vector.

Epipolar Constraint and Essential parameters

We observe that vectors x' ,T and Rx are coplanare because of the relation.

$$\mathrm{x}' = R\mathrm{x} + T$$

Also $(T \times R\mathrm{x})$ is orthogonal to both $R\mathrm{x}$ and T.Therefore we have

$$\mathrm{x}' \cdot (T \times R\mathrm{x}) = 0 \tag{2.8}$$

But $T \times R\mathrm{x} = [T]_{\times} R\mathrm{x}$.Therefore we have

$$\mathrm{x}' E \mathrm{x} = 0$$

where

$$E = \begin{bmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{bmatrix} = \begin{bmatrix} 0 & -T_3 & T_2 \\ T_3 & 0 & -T_1 \\ -T_2 & T_1 & 0 \end{bmatrix} R \tag{2.9}$$

The elements of matrix E are called essential parameters.The nine essential parameters are not independent because E is the product of a skew-symmetric

and rotation matrix.Dividing both sides of eqn(2.8) by zz' we get

$$\mathbf{X}'\hat{E}\mathbf{X} = 0 \tag{2.10}$$

which is a linear ,homogeneous equation in terms of the nine essential parameters. Equation(2.10) is called as the **Epipolar** constraint for 3D motion estimation ,and is the basis of linear estimation methods.

Decomposition of the E-Matrix

Theoretically from equation(2.8) the E Matrix can be expressed as

$$E = [e_1 \;\; e_2 \;\; e_3] = [k\hat{T} \times r_1 \;\; k\hat{T} \times r_2 \;\; k\hat{T} \times r_3] \tag{2.11}$$

where r_i are the columns of R and $k\hat{T} = T$.In the following we discuss methods to recover R and$\hat{T}$ given E computed from noise-free and noisy point correspondence data,respectively.

The E Matrix method in noise free environment

A detailed description of the algorithm can be found in [6] [9]

1. We have

   $$T \times \hat{T} = 0$$

   where $\hat{T}$ is a unit vector along T.From equations(2.1) , (2.2) and (2.3) it follows that

   $$\frac{z'}{\|T\|}X' = \frac{z}{\|T\|}RX + \frac{T}{\|T\|} \tag{2.12}$$

   Taking cross product of both sides of equation(2.12) by $k\hat{T}$ ($k\hat{T} = T$) we get

$$\frac{z'}{\|T\|}k\hat{T} \times X' = \frac{z}{\|T\|}k\hat{T} \times RX \tag{2.13}$$

PreMultiplying both sides of equation(2.13) byX'^T we get

$$X'^T[k\hat{T}]_\times RX = 0 \tag{2.14}$$

where we define

$$E = [k\hat{T}]_\times R \tag{2.15}$$

It follows that,

$$E = [\ e_1 \mid e_2 \mid e_3\ ] = [\ k\hat{T} \times r_1 \mid k\hat{T} \times r_2 \mid k\hat{T} \times r_3\ ] \tag{2.16}$$

where

$$R = [\ r_1 \mid r_2 \mid r_3\ ]$$

It follows that

2. From equation(2.16) we observe that each column of E is orthogonal to T.Then the unit vector along T ie.., $\hat{T}$ can be obtained as,

$$\hat{T} = \pm\frac{e_i \times e_j}{\|e_i \times e_j\|} \quad i \neq j \tag{2.17}$$

It follows that

$$k^2 = 0.5(e_1.e_1 + e_2.e_2 + e_3.e_3) \tag{2.18}$$

But the above also holds true if the direction of $\hat{T}$ is reversed.To resolve the confusion and find the correct direction the following test is done.From equation(2.13) and equation(2.16) it follows that

$$\frac{z'}{\|T\|}\hat{T} \times X' = \frac{z}{\|T\|k}EX$$

since $z, z' > 0$ we have $T \times X'$ and EX are of the same sign.Therefore

$$[\hat{T} \times X'].EX$$

is always positive and

$$\sum[\hat{T} \times X'] \cdot EX > 0 \tag{2.19}$$

if direction of $\hat{T}$ is correct.If the above summation is negative,reverse $\hat{T}$

Now that $\hat{T}$ is determined correctly,we proceed to determine R. Using the vector identity

$$A \times (B \times C) = (A.C)B - (A.B)C$$

we have

$$e_1 \times e_2 = e_1 \times (k\hat{T} \times r_2)$$

$$= [(k\hat{T} \times r_1).r_2]k\hat{T} - [(k\hat{T} \times r_1).k\hat{T}]r_2$$

The second term being zero it simplifies to

$$[r_1 \times r_2.k\hat{T}]k\hat{T} = (r_3.k\hat{T})k\hat{T}$$

We have made use of the fact that for any rotational matrix R, $RR^T = I$ ie..., $r_1 \times r_2 = r_3$ and so on .Therefore we have

$$e_1 \times e_2 = k\hat{T}(k\hat{T}.r_3) \tag{2.20}$$

$$e_2 \times e_3 = k\hat{T}(k\hat{T}.r_1) \tag{2.21}$$

$$e_3 \times e_1 = k\hat{T}(k\hat{T}.r_2) \tag{2.22}$$

Any unknown vector can be found if we know it's dot and cross product with a known vector.Refer figure 2.4. For example,

$$r_1 = (\hat{T}.r_1)\hat{T} + (\hat{T} \times r_1) \times \hat{T}$$

It follows from equations(2.16) and (2.21) that

$$r_1 = [\frac{1}{k^2}\hat{T}.(e_2 \times e_3)]\hat{T} + \frac{1}{k}(e_1 \times \hat{T}) \tag{2.23}$$

$$r_2 = [\frac{1}{k^2}\hat{T}.(e_3 \times e_1)]\hat{T} + \frac{1}{k}(e_2 \times \hat{T}) \tag{2.24}$$

$$r_3 = [\frac{1}{k^2}\hat{T}.(e_1 \times e_2)]\hat{T} + \frac{1}{k}(e_3 \times \hat{T}) \tag{2.25}$$

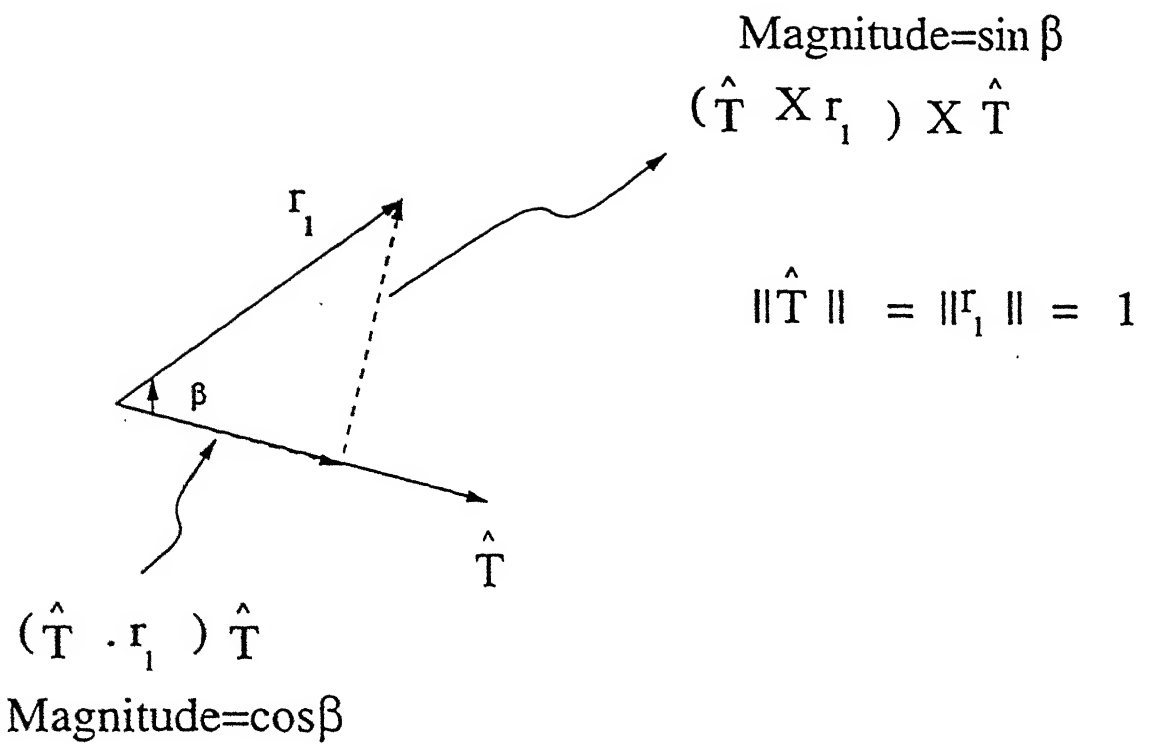


Figure 2.4: Orthogonal projection of r_1 onto $\hat{T}$

E Matrix Method in Noisy Environments

The **E Matrix** method in noisy emvironments is given below .[6] gives this algorithm in detail alongwith **Error Analysis** using **Vector Algebra** while [5] gives the same results using **Matrix Algebra**.The constraints in [5] are overcome in [6]. [9] also discusses the **E Matrix** method for noise-free and noisy environments.

Since the point correspondences are contaminated by noise,we may obtain different estimates of unit translation vector by using different combinations of e_i in equation(2.17) and the rotation matrix obtained by equations(2.23), (2.24) and (2.25) may no longer satisfy the properties of a rotation matrix.To

address these problems the algorithm is modified suitably to work in noisy environments.

There is an inherent deficiency in all model based approaches.The depth parameter,which says how far the model is located from the image plane,can't be determined.This is because if the model was located twice as far and was twice as big and moved twice as fast as the present model,the projected images would still be the same.Because of the inherent deficiency we assume $T = \hat{T}$.

1. *Determine E*

   Let $X_i = (u_i, v_i, 1)^T$ and $X_i' = (u_i', v_i', 1)^T$,$i = 1, 2, 3, ...n$ be the $n(n \geq 8)$ point correspondences. See equations(2.14) and (2.16).Out of the n point correspondences each point correspondence gives

   $$X_i'^T E X_i = 0 \qquad (2.26)$$

   Hence n points will give n equations of the form of equation(2.26).In other words,in noise free environments , equation (2.26) can be written in the form,

   $$Ah = 0$$

   where

   $$h = (h1, h2, h3, h4, h5, h6, h7, h8, h9)^T \qquad (2.27)$$

$$A = \begin{bmatrix} u_1u'_1 & u_1v'_1 & u1 & v_1u'_1 & v_1v'_1 & v_1 & u'_1 & v'_1 & 1 \\ u_2u'_2 & u_2v'_2 & u2 & v_2u'_2 & v_2v'_2 & v_2 & u'_2 & v'_2 & 1 \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ u_nu'_n & u_nv'_n & un & v_nu'_n & v_nv'_n & v_n & u'_n & v'_n & 1 \end{bmatrix} \tag{2.28}$$

In noisy environments we find h such that

$$\|Ah\| = minimum \tag{2.29}$$

The solution of h for minimum norm is the unit eigen vector of A^TA associated with the smallest eigen value. The depth parameter cannot be solved-the inherent deficiency in model based approach-so we solve for unit translation vector.Assume $k = 1$ we get

$$E = [\hat{T}]_\times R$$

Any Multiple of E will still be a solution of equation(2.14).Also $\|E\|^2 = 2$ (follows from the above definition).Norm of $\|E\|^2$ is $[\hat{T}]_\times RR^T[\hat{T}]_\times^T$.Since $RR^T = I$,the identity matrix and $[\hat{T}]_\times[\hat{T}]_\times^T = 2$ it follows that $\|E\|^2 = 2$.Therefore

$$E = [e_1\ e_2\ e_3] = \sqrt{2}\begin{bmatrix} h_1 & h_4 & h_7 \\ h_2 & h_5 & h_8 \\ h_3 & h_6 & h_9 \end{bmatrix} \tag{2.30}$$

This is the solution of **E** in noisy environment.Having solved **E** we now proceed to solve for $\hat{T}$,the unit translation vector

2. *Determine unit translation Vector*

Each column of E is orthogonal to $\hat{T}$ as seen from equation(2.16).Therefore.in noise free environments we have

$$e_i \times \hat{T} = 0 \quad i = 1, 2, 3$$

or in other words

$$E^T \hat{T} = 0$$

In noisy environmemts.as is always the case,we try to find $\hat{T}$ such that

$$\|E^T \hat{T}\| = minimum \tag{2.31}$$

The solution of $\hat{T}$ for minimum norm is the unit eigenvector of EE^T associated with it's smallest eigen value.But the above holds true for $-\hat{T}$ also .In order to find the correct direction of $\hat{T}$ we evaluate equation (2.19).If the summation

$$\sum[\hat{T} \times X'] \cdot EX > 0$$

retain $\hat{T}$ else

$$\hat{T} \leftarrow -\hat{T}$$

Though the summation is over all n points,usually a few points will suffice

3. *Determine rotation matrix .R*

Without noise we have

$$E = [\hat{T}]_\times R \tag{2.32}$$

which is

$$R^T[-\hat{T}]_\times = E^T \tag{2.33}$$

In noisy environments we find rotation matrix R such that

$$\|R^T[-\hat{T}]_\times - E^T\| = minimum \tag{2.34}$$

subject to R being a rotation matrix

Yet another way of finding the rotation matrix R is as follows. Refer equations (2.23), (2.24) and (2.25) .In Matrix form ($k = 1$) we have

$$R = W = [e_1 \times \hat{T} + e_2 \times e_3 \qquad e_2 \times \hat{T} + e_3 \times e_1 \qquad e_3 \times \hat{T} + e_1 \times e_2] \quad (2.35)$$

Let W denote an estimate of the rotation matrix where w_1 w_2 w_3 are the columns of W obtained from equations(2.23), (2.24) and (2.25).This is equal to the rotation matrix in noise-free environment.In noisy environments we find a rotation matrix R such that

$$\|R - W\| = minimum \quad (2.36)$$

subject to R being a rotation matrix.

It is here that we make use of **Quaternions** [17] [14] . Rotiation can be specified by a rotation matrix R or a quaternion Q.The relation is one-to-one.Both cases specified above are of the form where we have to find R such that

$$\|RC - D\| = minimum \quad (2.37)$$

subject to R being a rotation matrix

R is found ,using **Quaternions** ,as follows.It is shown in the appendix of [6] that

$$\|RC - D\|^2 = q^T B q \quad (2.38)$$

$$B = \sum_{i=1}^{3} B_i^T B_i \tag{2.39}$$

$$B_i = \begin{bmatrix} 0 & (C_i - D_i)^T \\ D_i - C_i & [D_i + C_i]_\times \end{bmatrix} \tag{2.40}$$

where $q = (q_0\ q_1\ q_2\ q_3)^T$ is the quaternion corresponding to rotation matrix R and C_i D_i $i = 1, 2, 3$ are the columns of C and D respectively The problem of finding R such that

$$\|RC - D\|^2 = minimum$$

is now reduced to finding q such that $q^T Bq$ is a minimum.q is now given as the unit eigen vector of B associated with it's smallest eigen value.

The relation between rotation matrix R and it's corresponding quaternion q is

$$R = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_2q_1 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{2.41}$$

4. *Check for correctness of $\hat{T}$ using R*

Let α be a small value($= 0$ in noise free environment)If

$$\|X_i' \times RX_i'\| / \|X_i'\| \|X_i\| \leq \alpha \quad 1 \leq i \leq n$$

we conclude $T \approx 0$,else evaluate

$$\sum_i (\hat{T} \times X_i') \cdot (X_i' \times RX_i) > 0 \tag{2.42}$$

If the above isn't true $\hat{T} \leftarrow -\hat{T}$

5. *Estimation of relative depths in case $T \neq 0$*

For $1 \leq i \lesssim n$ estimate the relative depth as ,

$$Z_i = (\frac{z_i'}{\|T\|} \quad \frac{z_i}{\|T\|}) = (\bar{z_i'} \quad \bar{z_i}) \tag{2.43}$$

Refer equation(2.12) we have

$$(X_i' \quad - RX_i)Z_i \quad - \quad \hat{T} = 0 \qquad \text{noise free environment}$$

Therefore ,in noisy environments, we find Z_i such that

$$\|(X_i' \quad - RX_i)Z_i \quad - \quad \hat{T}\| = minimum \tag{2.44}$$

The problem is to find x(2 x 1) vector such that

$$\|Ax - B\| = minimum$$

where A is a (3 x 2) and B is a (3 x 1) vector.There are 2 ways of minimizing the norm.They are

(a) **Singular Value Decomposition Method:**

The **Singular Value Decomposition** [15] of A is

$$A = UWV^T$$

where U(3 x 2)column orthogonal matrix,W(2 x 2)is a diagonal matrix with positive or zero elements(the singular values) and V(2 x 2)orthogonal matrix.x is now given as

$$x = VW^{-1}U^TB \tag{2.45}$$

(b) **Vector Algebra:**

In this method the norm is the distance between the vector B and it's projection on a plane passing thru' 2 vectors(These vectors are the first and second columns of A).

The corrected relative 3D position of point i is (scaled by $\|T\|^{-1}$since we are assuming unit translation vector)

$$\bar{x}'_i = (R(\bar{z}_i X_i) + \hat{T} + \bar{z}'_i X'_i)/2 \tag{2.46}$$

It's relative 3D position before rotation is

$$\bar{x}_i = R^{-1}(\bar{x}'_i - \hat{T}) \tag{2.47}$$

Refer [6] The 3D points before and after motion are back projected onto the image plane to give a new set of point correspondences as output.

2.3 A Matrix Method

Refer figure 2.3.The **A Matrix** method estimates the motion parameters and dimensions of a rigid planar patch.Refer [2, 3, 4]for a detailed discussion.It defines two transformations which relate the image point coordinates before and after motion.

2.3.1 The deformable(Generalized)block matching or A Matrix method

Let

$\mathbf{x} = (x, y, z)^T$ denote the 3D coordinates of a point before motion(time t1)

$\mathbf{x}' = (x', y', z')^T$ denote the 3D coordinates of the same point after motion(time t2)

$\mathbf{X} = (u, v, 1)^T = (\frac{x}{z}, \frac{y}{z}, 1)^T$ denote a vector of image plane coordinates at time t1

$\mathbf{X'} = (u', v', 1)^T = (\frac{x'}{z'}, \frac{y'}{z'}, 1)^T$ denote a vector of image plane coordinates at time t2

Refer [9, pages 109-111].From figure 2.3 and equations(2.1), (2.2) and (2.3) we get the following equations

$$u = F\frac{x}{z} \qquad v = F\frac{y}{z} \tag{2.48}$$

$$u' = F\frac{x'}{z'} \qquad v' = F\frac{y'}{z'} \tag{2.49}$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} \tag{2.50}$$

we have expanded matrices R and T respectively.Further we assume that all point lie on a planar patch whose equation is

$$ax + by + cz = 1 \tag{2.51}$$

From equations (2.49), (2.50) and (2.51) we get the following identities

$$u' = \frac{a_1 u + a_2 v + a_3}{a_7 u + a_8 v + 1} \equiv T_1(u, v) \tag{2.52}$$

$$v' = \frac{a_4 u + a_5 v + a_6}{a_7 u + a_8 v + 1} \equiv T_2(u, v) \tag{2.53}$$

$$A = k^{-1}\left\{R + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} [a\ b\ c]\right\} \tag{2.54}$$

where

$$A = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & 1 \end{bmatrix} \qquad k = r_{33} + c.\Delta z \qquad T = \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix}$$

$a_1 \ldots a_8$ are the eight **Pureparameters** and $T_1()$ $T_2()$are the 2 **transformations** which relate the Image Plane coordinates in Image Frame(N) to Image Frame(N-1).

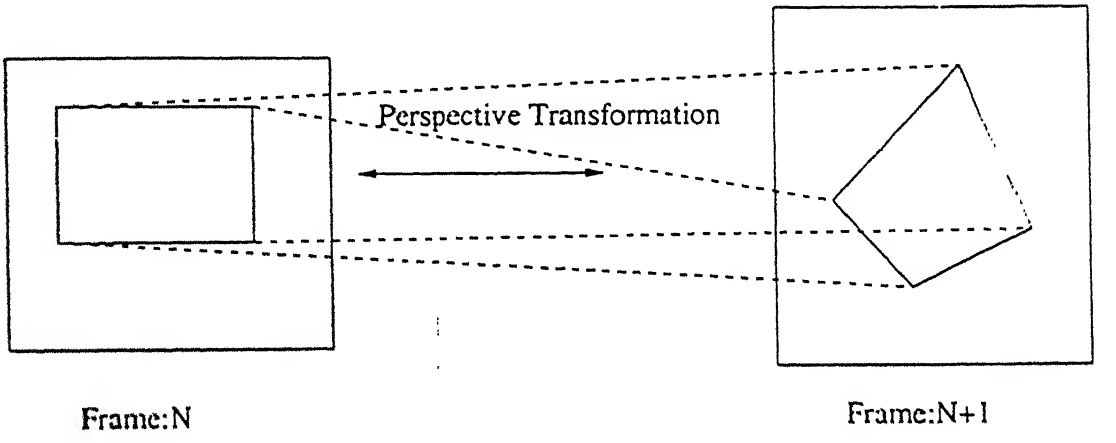


Figure 2.5: Generalized(deformable)block matching

In the above figure the corner points of the two quadrilaterals are the point correspondences.From these four point correspondences the pure parameteres can be found out.The **generalized (deformable)block matching** then consists of the following steps

- Divide the image into disjoint blocks (square shaped).For each square.estimate the point correspondence of it's corners using the standard block matching(3 step heirarchical search)

- Once the point correspondences of the corners of the square are computed,find out the eight **Pureparameters** using the above equations
- use these 8 values in the above two transformations to find the point correspondences of all points inside the square.For instance,in figure 2.5,all points inside the square on the left are displaced to an unique point inside the quadrilateral on the right
- perform the motion compensation,described in the preceding steps,on all squares inthe image

In references [2, 3, 4] algorithms are described wherein the motion and geometric parameters of a planar patch can be recovered once the pureparameters are known We implemented the algorithm in [3] to recover the motion parameters from the pureparameters .A planar patch was given a known translation and rotation and from the point correspondences ,the pureparametres were estimated.From this we could recover the motion parameters very accurately.

But on perturbation of the point correspondences(even by a margin of .1%)the variation of the recovered motion parameters was quite high.Since,in practice,noise can't be dispensed with,we had to switch over to the **E Matrix Method**which was specially modified to operate in noisy environments.Morever,the limitation of a planar patch was also overcome since the E matrix holds good in non-planar cases also.

2.4 The New Motion Estimation Algorithm

In this section,we describe in detail the motion compensation followed in this thesis.Before we transmit the first frame we separate the silhoutte of the person's face and determine the rectangle which completely encloses the face and shoulder region.This is called the "Mask" as it is this region which will be transmitted.The remaining region is the background which has little motion and we retain the same background for all images.The entire method of motion compensation is enumerated below

1. To begin with we are having two frames ,the N^{th} frame and the $(N-1)^{th}$ frame.We will motion compensate the $(N-1)^{th}$ frame to produce a faithful reproduction of the N^{th} frame.

2. Divide the rectangular region or the mask in the $(N-1)^{th}$ frame into disjoint square blocks of size $N \times N$ .Estimate the 2D displacement of the centre of each block using the standard block matching method already described.

3. Now we have the point correspondences of the centres of the square blocks.This is the input to the **E Matrix** Method explained in sec 2.2.3 .The E Matrix calculates the 3D coordinates of these points and these are projected back on the image plane to give a new set of point correspondences.

4. Let pixel (m,n) in the $(N-1)^{th}$ frame and pixel $(m+i,n+j)$ in the N^{th} frame be one such point correspondence.Then the entire block of pixels whose centre is the (m,n) pixel in the $(N-1)^{th}$ frame is displaced by

(i, j) in the reconstructed N^{th} frame before transmitting.This is referred to as Motion compensated $(N-1)^{th}$ frame

Mathematically speaking

$$\mathcal{F}(a,b) = \mathcal{G}(a+i, b+j) \qquad (a,b) \in \mathcal{B}$$

where $\mathcal{B}$ is the block of pixels in $(N-1)^{th}$ frame with centre (m,n) and $\mathcal{F}(a,b)$ is the value of pixel $(a.b)$ in the $(N-1)^{th}$ frame and $\mathcal{G}(a,b)$ is the value of pixel (a,b) in the reconstructed N^{th} frame .This is referred to as Motion compensated $(N-1)^{th}$ frame

We show the entire process of motion estimation and compensation in figure 2.6

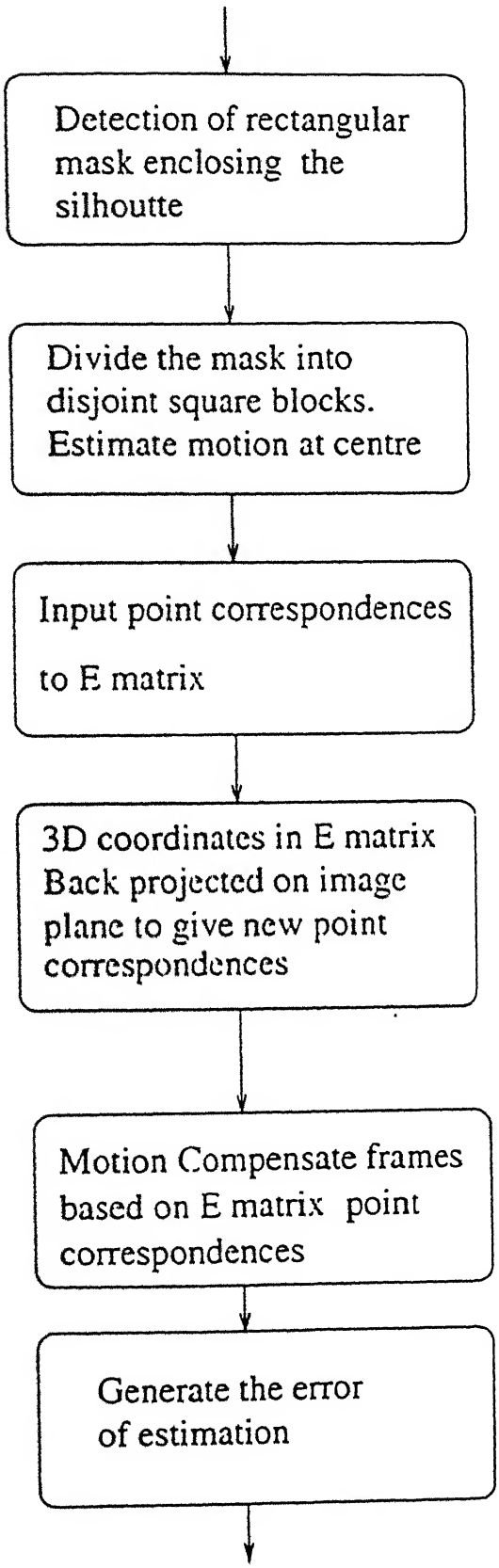


Figure 2.6: Flowchart of motion estimation and compensation

Chapter 3

Video Encoder and Decoder

3.1 Introduction

At the transmitting side,the error between each incoming frame and the motion compensated reconstructed frame is fed as the input to the video encoder which transmits it along with motion parameters as serial bits at the desired rate..At the receiving side the video decoder receives the serial bits and decodes it to recover the error before doing some processing to reconstruct and display the images.

In this chapter we shall discuss in detail the various stages of the video encoder and decoder

The images can be classified into four categories

- I Picture: This image is directly encoded without any motion compen-

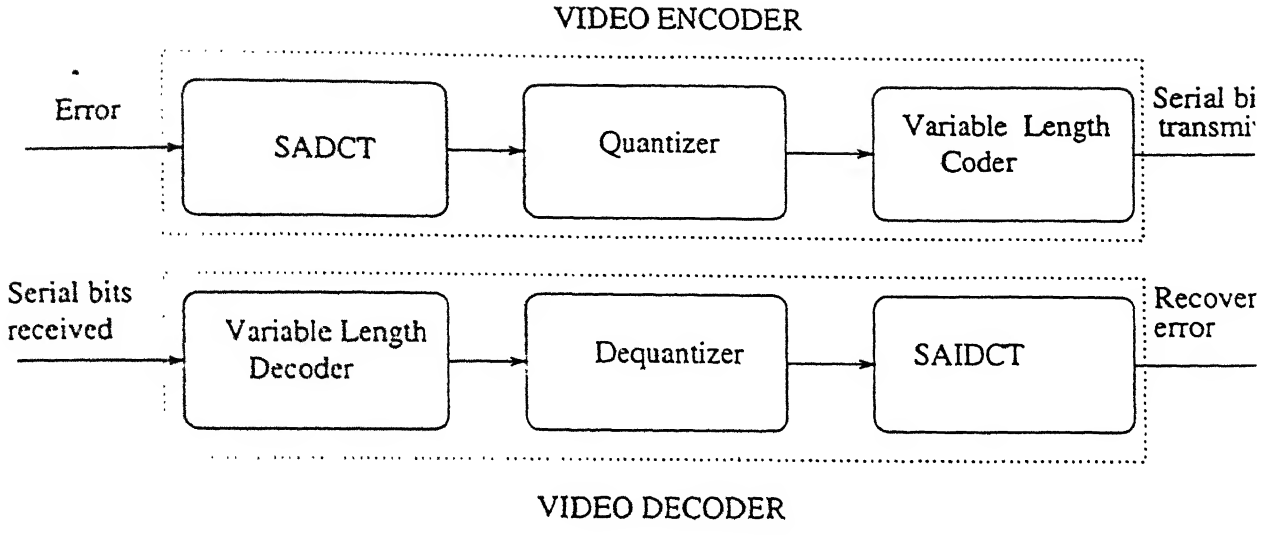


Figure 3.1: Block diagram of the video encoder and decoder

sation and estimation and provides a random access into the compressed data.It is also called as **Intra coding**

- P Picture: This picture is coded only after motion estimation and compensation.It is also called predictive coding or **Inter Coding** .This picture can be coded using a motion compensated I/P picture
- D Picture: is a special case of intra where only the DC coefficient is coded.
- B Picture: These images are coded using past/future images ie..., motion estimation/compensation is bidirectional.

In our case we do not use the D/B pictures.The first image is **Intra** coded while the next 10 images are **Inter** coded. In the subsequent sections only 8×8 blocks will be considered unless otherwise mentioned.

3.2 Shape Adaptive DCT

3.2.1 Features of SADCT

In practice we encounter arbitrarily shaped regions or contours of regions where the standard DCT is applied.Of late,a new type of DCT called the SADCT was developed to be used in such cases.Four such instances are shown in the figure below

This provides the means for generic coding of segmented video over a wide range of bit rates.After the SADCT the number of transform coefficients is the same as the number of non-zero pixels in the 8×8 block.Another advantage is that it is backward compatible with any video coder using standard DCT,because in the case of a 8×8 block having no zero pixels,the SADCT reducesto the Standard DCT.

The encouraging results of SADCT as against DCT(Rate-Distortion curve) reported in [13] prompted us to try out this method as the very purpose of this thesis was to reduce the bitrate without compromising on the image quality.In addition defects like "Mosquito effects" and "Blocking Artifacts"(that are present if DCT is used) do not occur in SADCT

3.2.2 The Algorithm

Fig(3.3(a))shows an arbitrarily shaped region inside a square block.The SADCT first relocates the values of the pixels in this region.Each column in this region is moved upwards to occupy the first few rows. For eg the 3 asterix-marked pixels

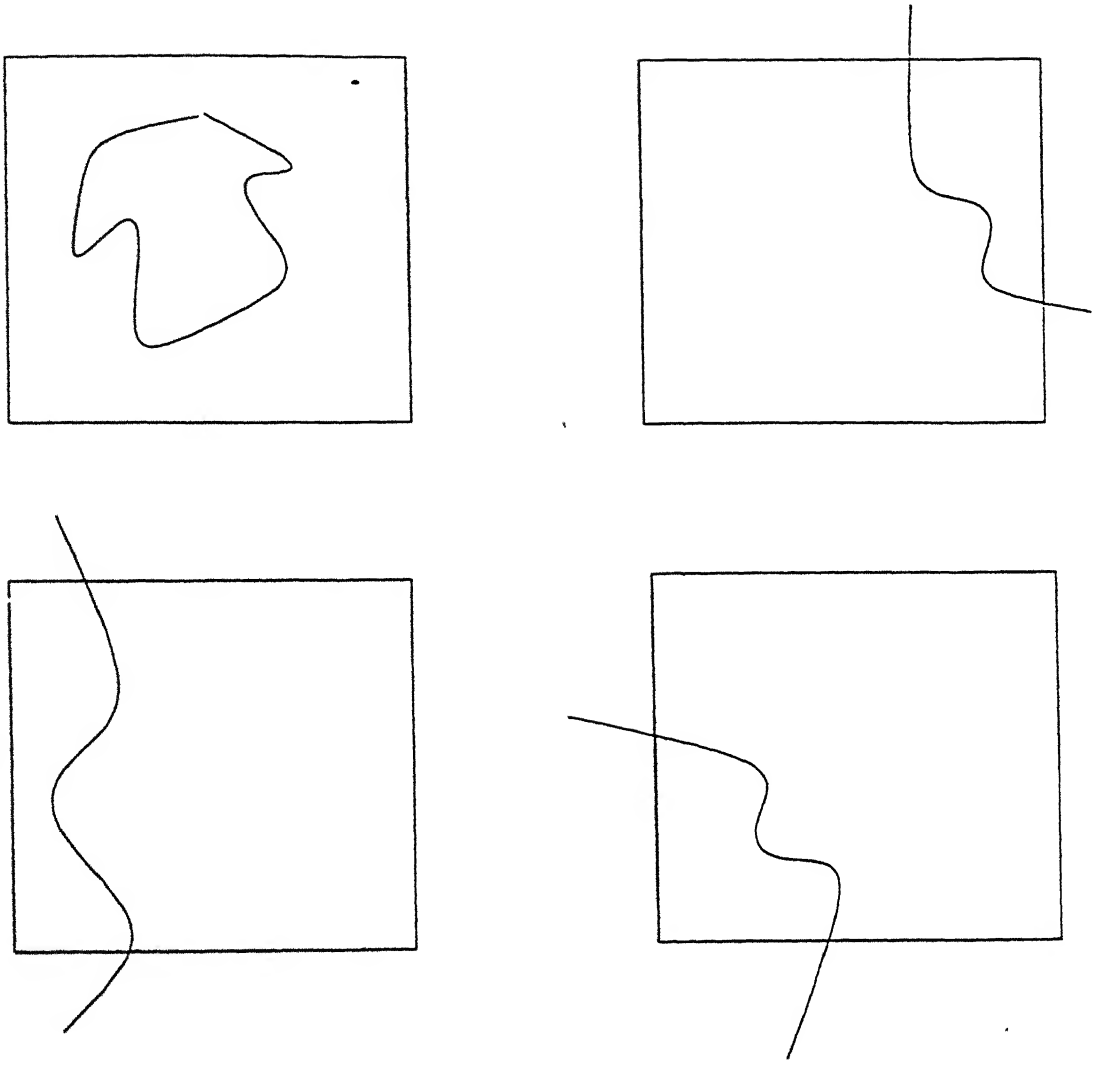


Figure 3.2: Cases where SADCT is recommended,the curved line denotes the contour and the square denotes the 8 × 8 block

in the 3^{rd} column of Fig(3.3(a))are moved upwards along the same column to occupy the first 3 rows.This is done for all the other columns as well.The result is Fig(3.3(b)).Now the standard 1D DCT-N is applied to each column,where N is the number of nonzero values in that column.DCT-3 is applied to the 3^{rd} column of Fig(3.3(b)) to give the 3^{rd} column of Fig(3.3(c)).This is repeated for all columns of Fig(3.3(b)).The result is Fig(3.3(c)).Each column in Fig(3.3(c))

is the transform of the corresponding column in Fig(3.3(b)).A similar relocation is done here as before except that it is in the horizontal direction All values are moved towards the first column.The result is Fig(3.3(d)). Now 1D DCT-N is applied to each row in Fig(3.3(d))(N is the number of nonzero values in that row)to give Fig(3.3(e)) which is the final set of SADCT coefficients of the arbitrarily shaped region.

We note that

- The number of SADCT transform coefficients is the same as the number of pixels in the arbitrarily shaped region
- Note that all the transform coefficients are located in the upper left hand corner.This will especially give low bitrates as zigzag scan is employed in coding

3.3 Quantization

The output of the SADCT is the input to the quantizer.The quantizer truncates the values of the transform coefficients to appropriate Levels and it is these levels that are encoded and transmitted as serial bits.The quantizer is defined as follows.

It is a function $Q(.)$ with a finite set of decision levels d_i and reconstruction levels r_i such that

$$Q(s) = r_{i-1} \quad s \in (d_{i-1}, d_i] \quad \|i\| = 1, \ldots 127 \tag{3.1}$$

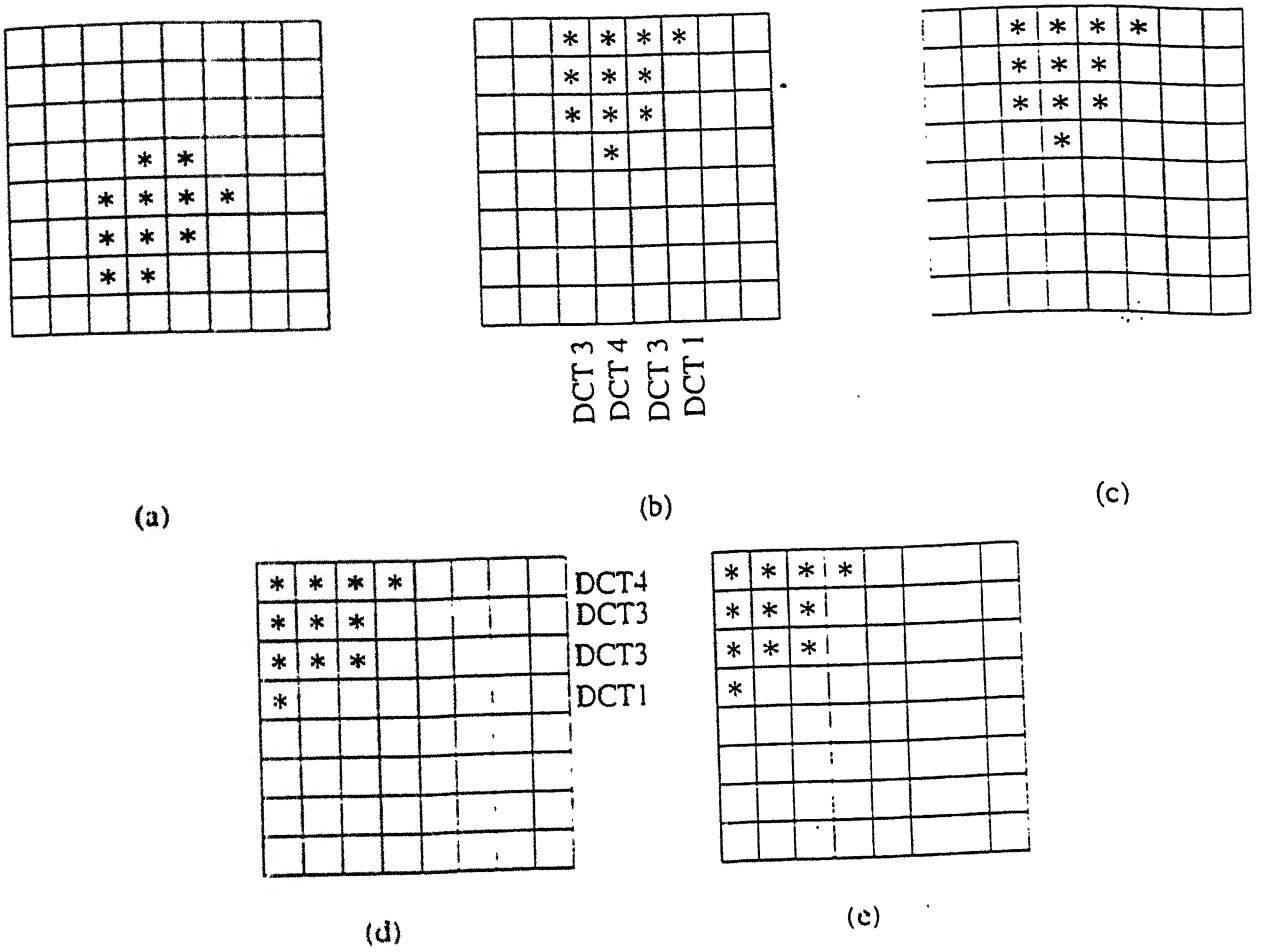


Figure 3.3: The various steps of the SADCT on a arbitrarily shaped region

The region between d_{-1} and d_1 is called as **Deadzone** and $d_i - d_{i-1}$ for $i \geq 2$ or $d_i - d_{i+1}$ for $i \leq -2$ is the **Stepsize**.The rate-distortion curve depends on the stepsize and deadzone.A comparative study of this is made in the next chapter with the help of examples.

We have followed the following logic to find reconstruction levels

- If $(s \geq d_{127})$ $\quad \mathcal{Q}(s) = 127$
- If $(s \leq d_{-127})$ $\quad \mathcal{Q}(s) = -127$

- If $(d_{-1} < s < d_1) \quad Q(s) = 0$
- If $s > d_1$ and $(d_{i-1} \leq s < d_i) \quad (i > 1) \quad Q(s) = i - 1$
- If $s < d_{-1}$ and $(d_i < s \leq d_{i+1}) \quad (i < -1) \quad Q(s) = i + 1$

Once these rules are known the Dequantizer can dequantize the levels.

3.4 Video Encoder and Decoder

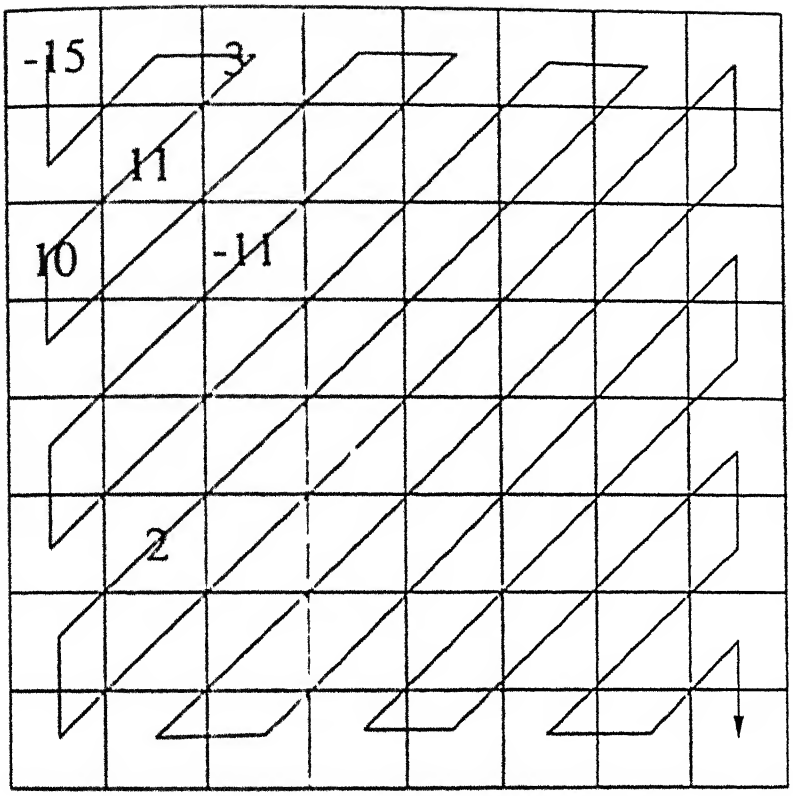
The VLC codewords are given in Tables (3.1) and (3.2) [16. page 480].Replace the last bit by 1 to get codeword for negative levels.

The Quantized transform coefficients (ie...,levels)are zigzag scanned to form a sequence of (Run,level) pairs.Beginning at the upper left hand corner we scan the the coefficients in zigzag fashion.The number of zero values passed before encountering the nonzero value is the Run count and the nonzero value is the Level count.Now the run is initialized to zero and we traverse further down the zigzag scan till we come across another nonzero value.Then we form the next (run,level)pair.This process is continued till either the zigzag scan ends in a non-zero value or there is a chain of zero values till the end after a nonzero value, in which case we transmit the "10" denoting EOB(End of Buffer).

For each (Run,Level) we transmit the corresonding codeword from the VLC table.The codewords for a negative level in a (Run,Level) can be got as follows.Find the codeword of (Run,Plevel) where level is the negative of Plevel.Replace the last bit of (Run,Plevel)(This is 0 for all positive levels)by 1.This is the codeword of (Run,Level)where Level is negative.There will be a

few (Run,Level)combinations with no entry in VLC table.In that case we transmit a 20bit codeword where the first 6 bits denote "ESCAPE" and the next 6 bits denote RUN.The last 8 bits denote LEVEL. In this case RUN,LEVEL are FLC coded. The motion parameters are transmitted by FLC(Fixed Length Codes)

The example shown in figure 3.4 shows the serial bits for the quantized transform coefficients.



Run-Level Pairs:(0,-15), (2,3),(0,11),(0,10),(6,-11),(13,2) EOB

Serial Bits:0000 0000 1011 11 0000 0010 110 0000 0001 000(
0000 0001 0011 0 0000 01 000110 11110101
0000 01 001101 00000010

Figure 3.4: Zigzag scan of coefficients.The (Run,Level)pairs and Serial Bits that are transmitted are also shown

Table 3.1: Table for VLC codewords

Run	Level	Codeword	Run	Level	Codeword
0	1	110	0	2	01000
0	3	001010	0	4	00001100
0	5	0010011100	0	6	001000010
0	7	000000010100	0	8	0000000111010
0	9	0000000110000	0	10	0000000100110
0	11	0000000100000	0	12	00000000110100
0	13	0000000000110010	0	14	00000000110000
0	15	0000000000101110	1	1	0110
1	2	0001100	1	3	001001010
1	4	0000000011000	1	5	0000000110110
1	6	0000000000101100	1	7	00000000101010
2	1	01010	2	2	00001000
2	3	0000000010110	2	4	0000000101000
2	5	0000000000101000	3	1	001110
3	2	0010001000	3	3	0000000111000
3	4	0000000000100110	4	1	001100
4	2	0000000011110	4	3	0000000100100
5	1	0001110	5	2	00000010010
5	3	0000000000100100	6	1	0001010
6	2	0000000111100	7	1	0001000
7	2	0000000101010	8	1	00001110
8	2	0000000100010	9	1	00001010

Table 3.2: Table for VLC codewords *(contd.)*

Run	Level	Codeword	Run	Level	Codeword
9	2	000000000100010	10	1	001001110
10	2	000000000100000	11	1	001000110
12	1	001000100	13	1	001000000
14	1	000000011100	15	1	000000011010
16	1	000000010000	17	1	0000000111110
18	1	0000000110100	19	1	0000000110010
20	1	0000000101110	21	1	0000000101100
22	1	000000000111110	23	1	000000000111100
24	1	000000000111010	25	1	000000000111000
26	1	000000000110110			
0	1	10(If First Coefficient)	0	1	110(If not first coefficient)
ESCAPE		000001	EOB		10

Chapter 4

Results and Discussions

4.1 Determination of Silhoutte and Isolation of moving regions

On receiving the zeroth frame we detect the silhoutte as shown in figure 4.1 [1] .The background and the region other than the head and shoulder remain the same as there is very little motion in these areas.As such we transmit only the moving regions .Motion is estimated using the E-matrix method and the error in this region is coded and transmitted.At the receiving side we attach the stationary part fig 4.1 (d) to each frame and recover the moving region from the received serial bits.

Once the silhoutte fig4.1 (b) is detected we select a rectangular window large enough to hold the silhouttes of the various frames.This rectangle serves as the mask and the motion is estimated only in this rectangular region in each

frame.We append the stationary content and the background to each frame on the receiving side

The rectangular region in each frame is divided into disjoint square blocks of size $N \times N$ where N is odd.The motion of the central pixel in each block is estimated using the E Matrix Method

Various values for N were tried out,starting from 3,5,...13 .The point to be noted here is that if N increases we have lesser number of square blocks.Consequently, motion has to be estimated at lesser number of points and motion parameters transmitted require lesser number of bits.But the quality of the image deteriorates.If N were decreased ,quality improves and bits for motion parameters also increases.So the question arises whether it is worth improving the image quality at the expense of bitrate.We found that the bits for motion parameters increased appreciably when window size $N \times N$ changed from 11×11 to 7×7,5×5 etc. The improvement in the quality of image reconstructed was only marginal.So by trial and error ,a tradeoff between quality and bitrate was attained.we fixed the window size to 11×11.

4.2 Motion Estimation and Compensation

The rectangular region of the $(N-1)^{th}$ frame is divided into 11×11 square blocks and the motion of the central pixel of each of the blocks is estimated .Now we have a set of point correspondences which is the input to the E matrix method discussed in the chapter on motion estimation .The output of the E matrix method gives it's own point correspondences .Based on this we motion compensate the $(N-1)^{th}$ frame to get a reconstructed N^{th} frame on

the transmitting side.The difference of the N^{th} frame and motion compensated $(N-1)^{th}$ frame is coded and transmitted to the receiving side.

As an example ,we have considered 10 frames of the standard "claire" image sequence downloaded from ftp:ipl.rpi.edu. Figures 4.2(a) and 4.2(b) denote the Zeroth Frame and the First Frame of the claire sequence. Figure 4.2(c) denotes the absolute error between the First frame and Motion Compensated Zeroth Frame. Figure 4.2(d) denotes the Motion Compensated Zeroth Frame or the reconstructed First frame on the transmitting side.

4.3 E matrix method

In this section we show the results of the E matrix method.The results are shown in table .The motion estimation was done between the Zeroth and first frame of the Claire sequence.Focal length F,which is equal to 100.was normalized to 1.

The value of the E matrix is

$$E = \begin{bmatrix} -0.121 & -0.816 & -0.527 \\ 0.851 & -0.0132 & 0.164 \\ 0.516 & -0.157 & 0.00723 \end{bmatrix}$$

Rotation Matrix R is

$$R = \begin{bmatrix} 0.998 & -0.0637 & -0.0192 \\ 0.0637 & 0.998 & -0.00253 \\ 0.0193 & 0.0013 & 1 \end{bmatrix}.$$

Table 4.1: Point correspondences of 2-D and 3-D motion estimation

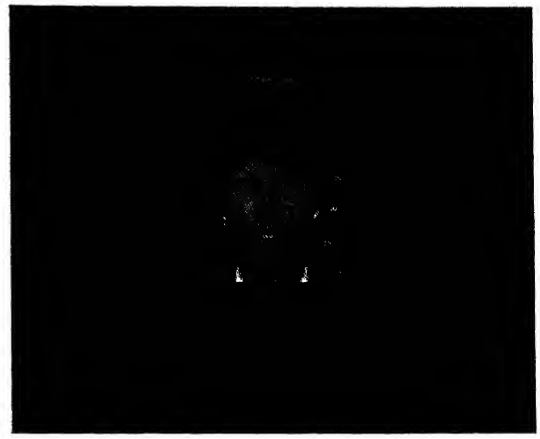
Position of pixel before motion		Position of the same pixel after 2D motion estimation		Position of the same pixel after 3D motion estimation	
column	row	column	row	column	row
162.000000	61.000000	162.000000	60.000000	162.8949697	64.481494
190.000000	61.000000	187.000000	70.000000	185.715696	68.513789
152.000000	89.000000	152.000000	86.000000	151.678828	87.131177
180.000000	89.000000	172.000000	85.000000	172.010477	84.960513
208.000000	89.000000	207.000000	94.000000	204.165228	91.589255
147.000000	117.000000	144.000000	126.000000	142.889362	125.706783
175.000000	117.000000	174.000000	115.000000	174.113784	114.932690
203.000000	117.000000	203.000000	115.000000	200.899847	114.915879
151.000000	145.000000	150.000000	140.000000	149.766593	139.916329
179.000000	145.000000	179.000000	144.000000	175.484484	133.784625
207.000000	145.000000	206.000000	144.000000	204.027951	143.120648
158.000000	173.000000	159.000000	176.000000	156.174616	179.099628
186.000000	173.000000	187.000000	171.000000	181.102081	156.693936
214.000000	173.000000	213.000000	174.000000	215.335418	181.751002

Translation Matrix is

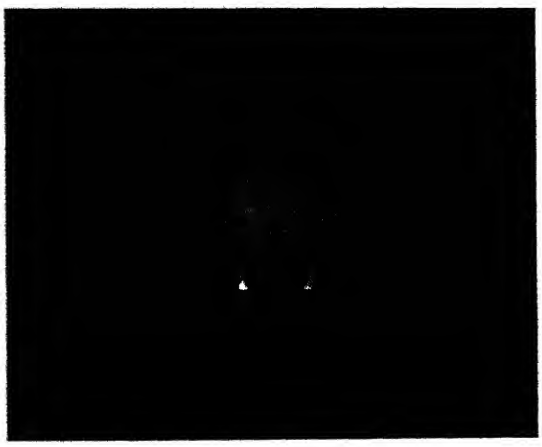
$$T = \begin{bmatrix} 0.153 \\ 0.528 \\ -0.835 \end{bmatrix}$$

The translation matrix is on the normalized scale. Root Mean Squared Error between 2D motion estimation and 3D motion estimation:5.914497

(a) The Zeroth Frame of Claire Sequence



(b) Silhoutte Detected

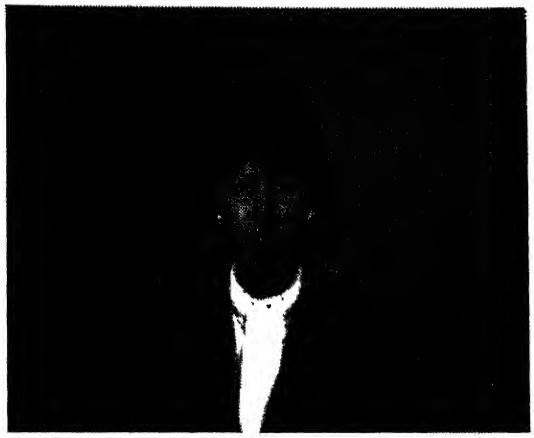


(c) Rectangle enclosing the Silhoutte

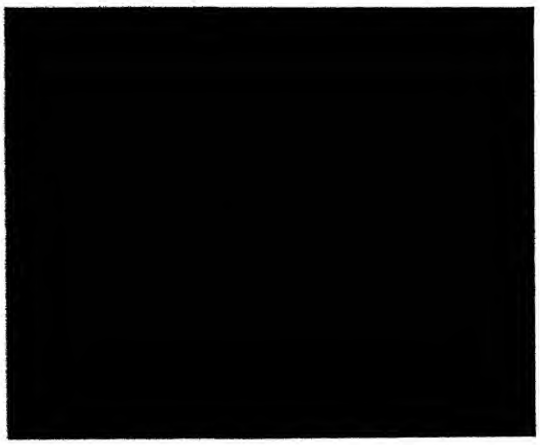
(d) Background and Stationary parts of the scene

Figure 4.1: Various steps of isolating the Head and Shoulder region

(a) Zeroth Image



(b) First Image



(c) Error of Estimating the First Image from the Zeroth Image

(d) Reconstructed First Image or Motion Compensated Zeroth Image

Figure 4.2: Comparison of Error Images and Motion Compensated Images

4.4 Video Encoder and Decoder

In this section, we discuss and show results pertaining to the working of the encoder and decoder.The row and column location refers to the upper left pixel of the 8 × 8 block in figure 4.2 (c).The rectangular region of the error image is divided into disjoint square blocks of size 8 × 8.Each block is coded and transmitted .This is recovered on the receiving side.

In the first example we consider a block which encloses part of the contour.

```
Row and column values of upper left pixel of 8 x 8 block is:85, 217
input to sadct
     2    21    35     2     0     0     0     0
     3    17    43    11     0     0     0     0
     3     9    37    25     1     0     0     0
     2     7    24    40     1     0     0     0
     3     4    14    50     9     0     0     0
     2     2     6    47    22     0     0     0
     6     7    39    58     5     2     0     0
     7     4    26    52    12     2     0     0
output of sadct
  115.1111  -0.2670 -94.9177  11.4444  32.7233 -13.8781   0.0000   0.0000
 -16.7266  21.4914  18.4543 -45.3828 -15.3465  30.1848   0.0000   0.0000
   1.1645  22.1956 -17.4650 -13.7870  17.8855   0.0000   0.0000   0.0000
  -8.2638  -1.5747  12.4401  -5.6644 -10.9737   0.0000   0.0000   0.0000
   1.6340  -8.4482  13.9079  -4.4778  -5.6323   0.0000   0.0000   0.0000
   2.0955   5.6785 -12.8928   4.6739   1.0649   0.0000   0.0000   0.0000
 -16.6355   8.0208   7.3230  -7.8137   0.0000   0.0000   0.0000   0.0000
```

```
   9.3749   -5.3241   -2.0378    5.8382    0.0000    0.0000    0.0000    0.0000

 stepsize:10.000000  dead_zone:29
quantized coeffs
 109.0000    0.0000  -89.0000    0.0000   29.0000    0.0000    0.0000    0.0000
   0.0000    0.0000    0.0000  -39.0000    0.0000   29.0000    0.0000    0.0000
   0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
   0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
   0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
   0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
   0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
   0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
quantized coeffs converted to levels
     9     0    -7     0     1     0     0     0
     0     0     0    -2     0     1     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0

 serial bits:0000000110000000001000010111110010001010010010010011101O
 number of bits:56
decoded levels
     9     0    -7     0     1     0     0     0
     0     0     0    -2     0     1     0     0
```

```
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
```

decoded quant coeffs

```
 109.0000    0.0000  -89.0000    0.0000   29.0000    0.0000    0.0000    0.0000
   0.0000    0.0000    0.0000  -39.0000    0.0000   29.0000    0.0000    0.0000
   0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
   0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
   0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
   0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
   0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
   0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
```

output of sa-idct

```
  -2.3585   10.2323   43.4438   16.1833    0.0000    0.0000    0.0000    0.0000
  -1.6092    9.9683   41.3687   18.2584    0.0000    0.0000    0.0000    0.0000
  -0.2248    9.4806   37.5344   22.0926    6.7909    0.0000    0.0000    0.0000
   1.5840    8.8434   32.5248   27.1023    7.2485    0.0000    0.0000    0.0000
   3.5419    8.1536   27.1023   32.5248    8.0409    0.0000    0.0000    0.0000
   5.3507    7.5164   22.0926   37.5344    8.9560    0.0000    0.0000    0.0000
   6.7352    7.0286   18.2584   41.3687    9.7485    6.1111    0.0000    0.0000
   7.4844    6.7647   16.1833   43.4438   10.2060   -0.9852    0.0000    0.0000
```

recovered output

```
    -2    10    43    16     0     0     0     0
    -2    10    41    18     0     0     0     0
```

```
 0     9     38    22     7     0     0     0
 2     9     33    27     7     0     0     0
 4     8     27    33     8     0     0     0
 5     8     22    38     9     0     0     0
 7     7     18    41    10     6     0     0
 7     7     16    43    10    -1     0     0
MSE: DCT input <-> SA-IDCT output: 71.775
```

This is in accordance to the rate-distortion curve shown in [13] .The mean square error is better if SADCT is used instead of DCT.

We show below the case of a block not located in model failure area(area where pixel mismatch or error is greater than 5).By reducing the deadzone and stepsize we can transmit this area with high efficiency and lesser bits.However ,we are using the same value of deadzone and stepsize for the entire rectangular region.By dynamically varying the deadzone and stepsize(using low values in non-model failure areas)we can get better quality images at lower bit rates.The same is shown below

Row,column values of upper left pixel of 8 x 8 block is:140, 194

input to sadct

2	3	6	6	5	1	1	1
4	0	3	2	4	2	0	1
2	1	2	1	2	0	0	1
3	1	1	1	2	1	1	1
2	0	0	0	3	3	2	0
3	2	2	2	0	4	3	0
6	4	3	3	1	1	2	0
4	1	1	2	0	1	1	1

output of sadct

17.6071	3.9623	-1.7345	1.8810	1.4167	1.0266	0.4478	1.7602
1.4616	-0.1250	-3.6931	-1.9532	0.3687	-1.0512	0.3273	1.7335
1.7036	3.0844	-1.5923	-2.3644	2.3181	0.5108	0.1200	0.3879
3.8930	0.9161	-2.0246	-1.0847	-1.3597	-0.4737	-0.9003	-0.8058
-1.1434	-1.2887	-0.3255	-0.5758	-0.4699	0.9331	0.0713	-0.9737
0.9493	2.1094	-0.9549	-0.9065	1.0503	-0.9952	0.4723	0.0000
-1.1481	-1.9870	-1.7835	-0.8570	0.0000	0.0000	0.0000	0.0000
-1.5433	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

stepsize:4.000000 dead_zone:8

quantized coeffs

16.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

```
  0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
  0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
  0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
quantized coeffs converted to levels
     3     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
 serial bits:00101010
 number of bits:8
decoded levels
     3     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
     0     0     0     0     0     0     0     0
 decoded quant coeffs
  16.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
   0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
```

```
   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
output of sa-idct
   2.0000   2.0000   2.0000   2.0000   2.0000   2.0000   2.0000   2.0000
   2.0000   0.0000   2.0000   2.0000   2.0000   2.0000   0.0000   2.0000
   2.0000   2.0000   2.0000   2.0000   2.0000   0.0000   0.0000   2.0000
   2.0000   2.0000   2.0000   2.0000   2.0000   2.0000   2.0000   2.0000
   2.0000   0.0000   0.0000   0.0000   2.0000   2.0000   2.0000   0.0000
   2.0000   2.0000   2.0000   2.0000   0.0000   2.0000   2.0000   0.0000
   2.0000   2.0000   2.0000   2.0000   2.0000   2.0000   2.0000   0.0000
   2.0000   2.0000   2.0000   2.0000   0.0000   2.0000   2.0000   2.0000
recovered output
     2     2     2     2     2     2     2     2
     2     0     2     2     2     2     0     2
     2     2     2     2     2     0     0     2
     2     2     2     2     2     2     2     2
     2     0     0     0     2     2     2     0
     2     2     2     2     0     2     2     0
     2     2     2     2     2     2     2     0
     2     2     2     2     0     2     2     2
MSE: DCT input <-> SA-IDCT output: 2.03846
```

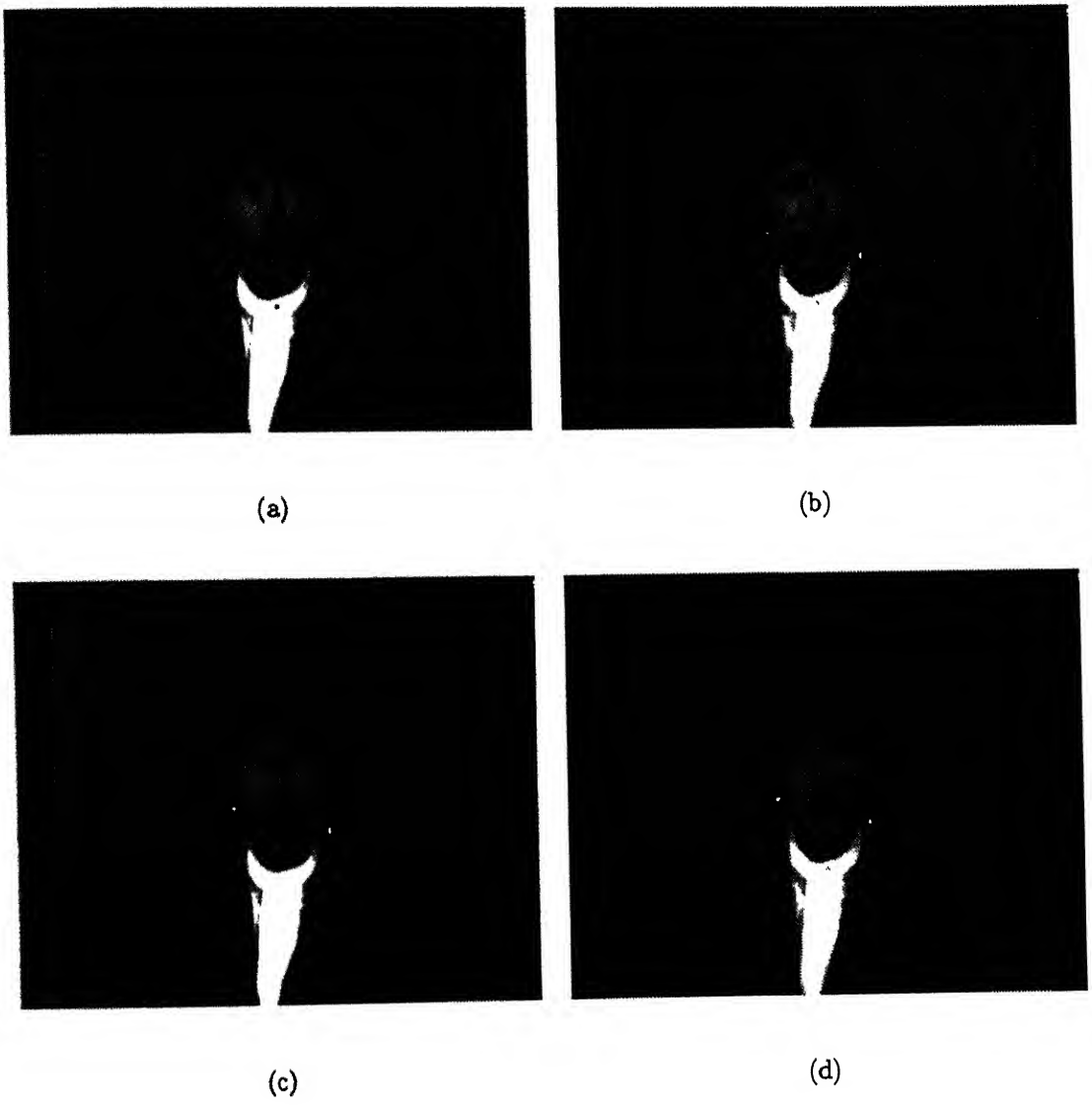
4.5 Comparison of images at various bitrates

In the following pages we compare the final reconstructed image on the receiving side with the original image.The comparison is done for images at 3 different bitrates.Also the algorithm was tested on the Claire,Miss America and Salesman sequence.

The reconstructed and original images are shown in figures 4.3, 4.4, 4.5, 4.6 and 4.7. (a) denotes the Original image in each of these figures
(b) denotes the reconstructed image on the receiving side in each of these figures.Bitrate is 63kbps.Deadzone is 25,Stepsize is 14
(c) denotes the reconstructed image on the receiving side in each of these figures.Bitrate is 61kbps.Deadzone is 29,Stepsize is 10
(d) denotes the reconstructed image on the receiving side in each of these figures.Bitrate is 57kbps.Deadzone is 31,Stepsize is 10

Figures 4.8,4.9 show the original and reconstructed images of the Miss America sequence.The values of bitrate,deadzone and stepsize are 68kbps.25 and 14 respectively.Figures 4.10,4.11 show the original and reconstructed imges of the Salesman sequence.The values of bitrate,deadzone and stepsize ar e 81kbps,25 and 14 respectively. For the Miss America and Salesman sequence the images on the left are the original ones while those on the right are the final reconstructed ones.

Comparison of Images at various bitrates

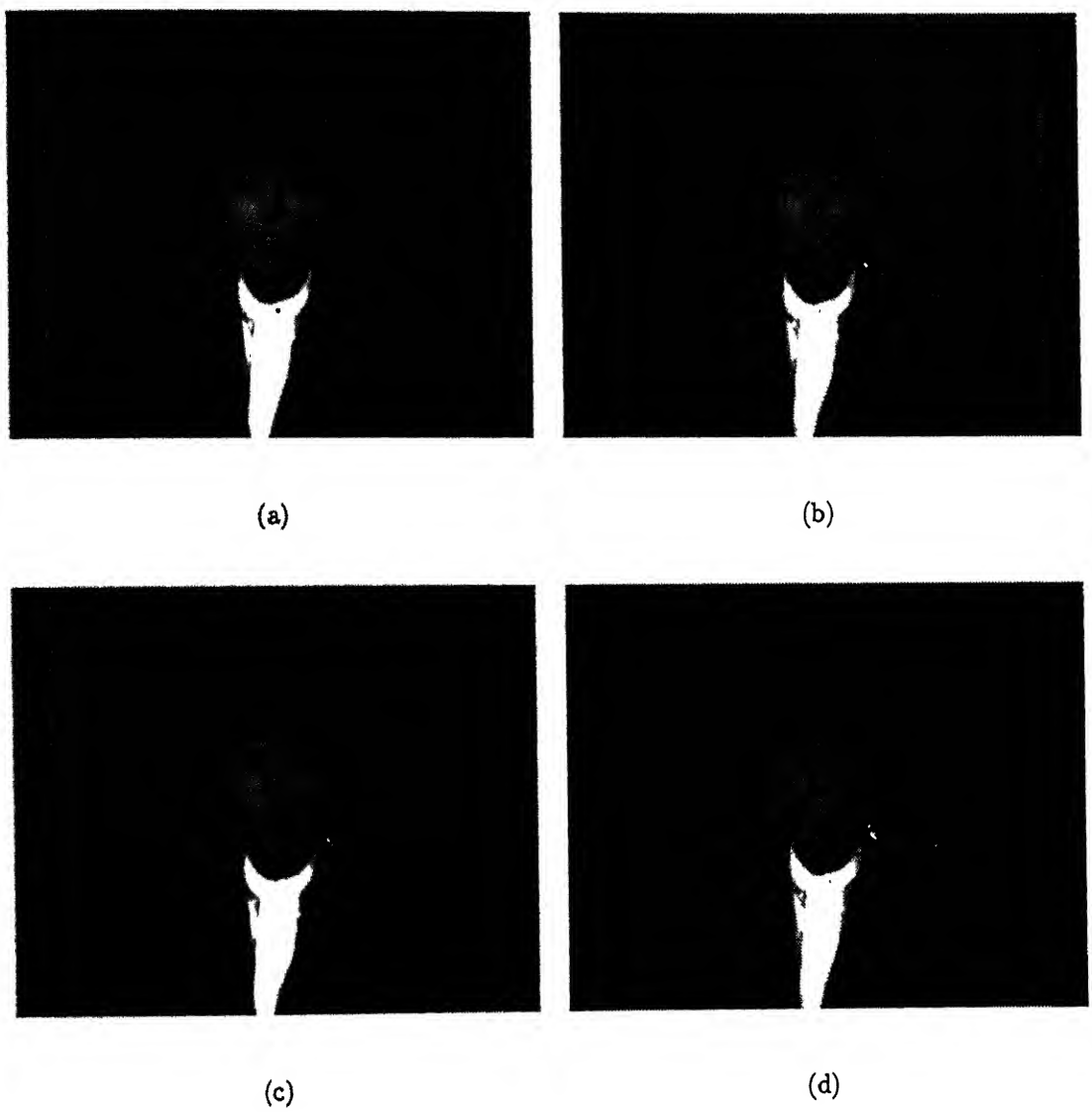


(a) (b)

(c) (d)

Figure 4.3: First original frame (a),reconstructed frames (b) ,(c),(d) at bitrates 63kbps,61kbps,57kbps respectively

Comparison of Images at various bitrates

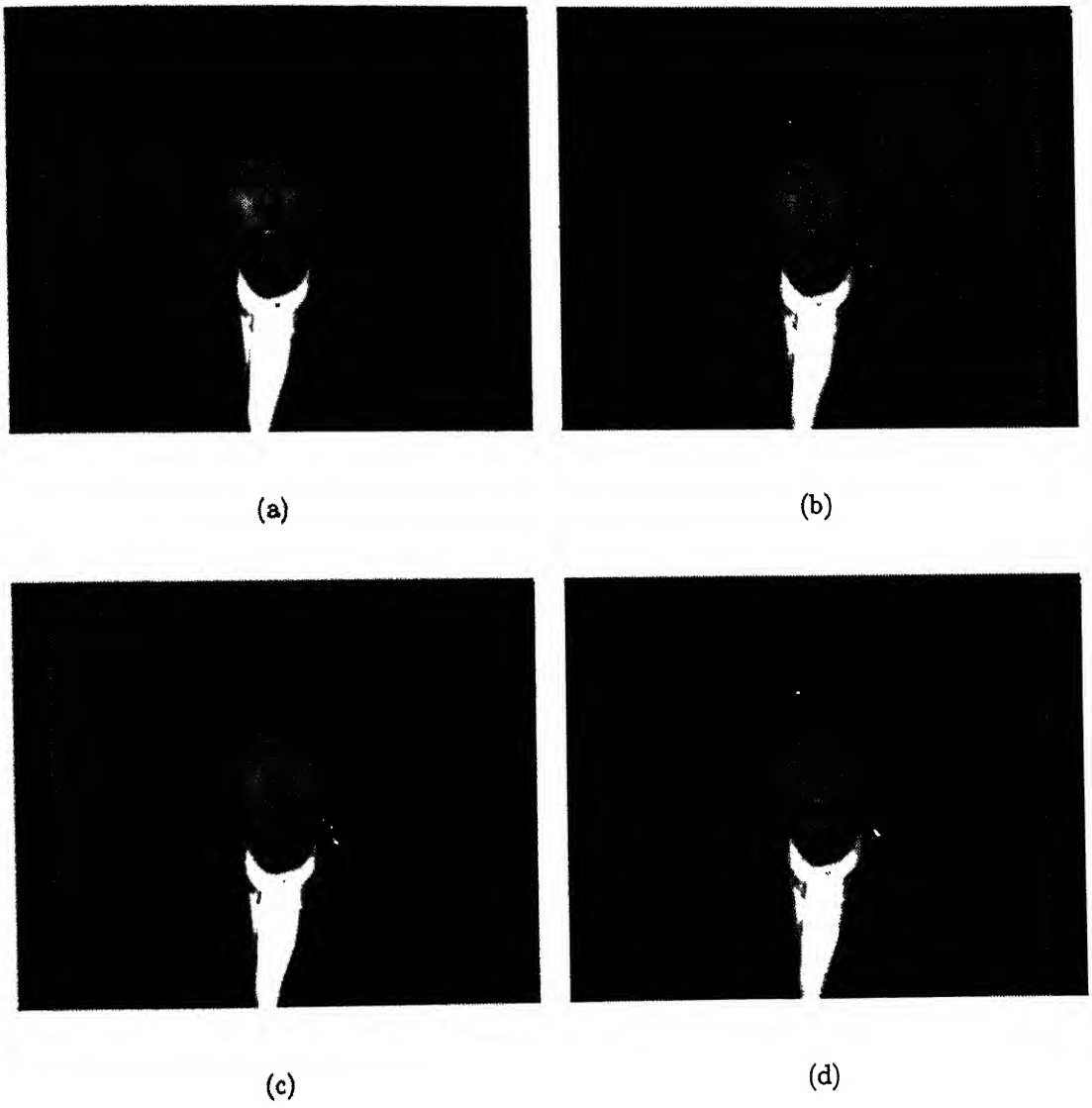


(a) (b)

(c) (d)

Figure 4.4: Third original frame (a),reconstructed frames (b) ,(c),(d) at b itrates 63kbps,61kbps,57kbps respectively

Comparison of Images at various bitrates

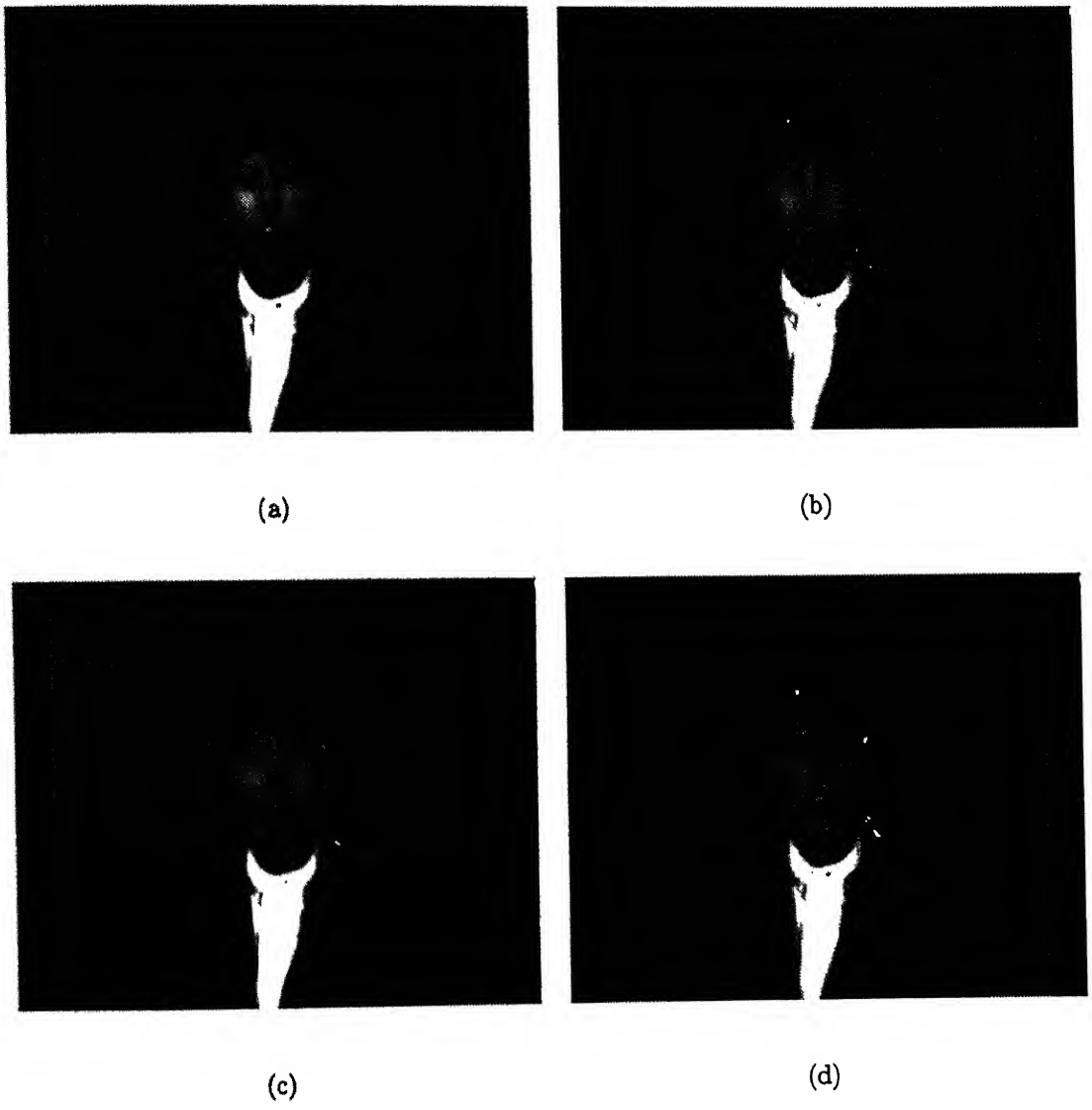


(a) (b)

(c) (d)

Figure 4.5: Fifth original frame (a),reconstructed frames (b) ,(c),(d) at b itrates 63kbps,61kbps,57kbps respectively

Comparison of Images at various bitrates

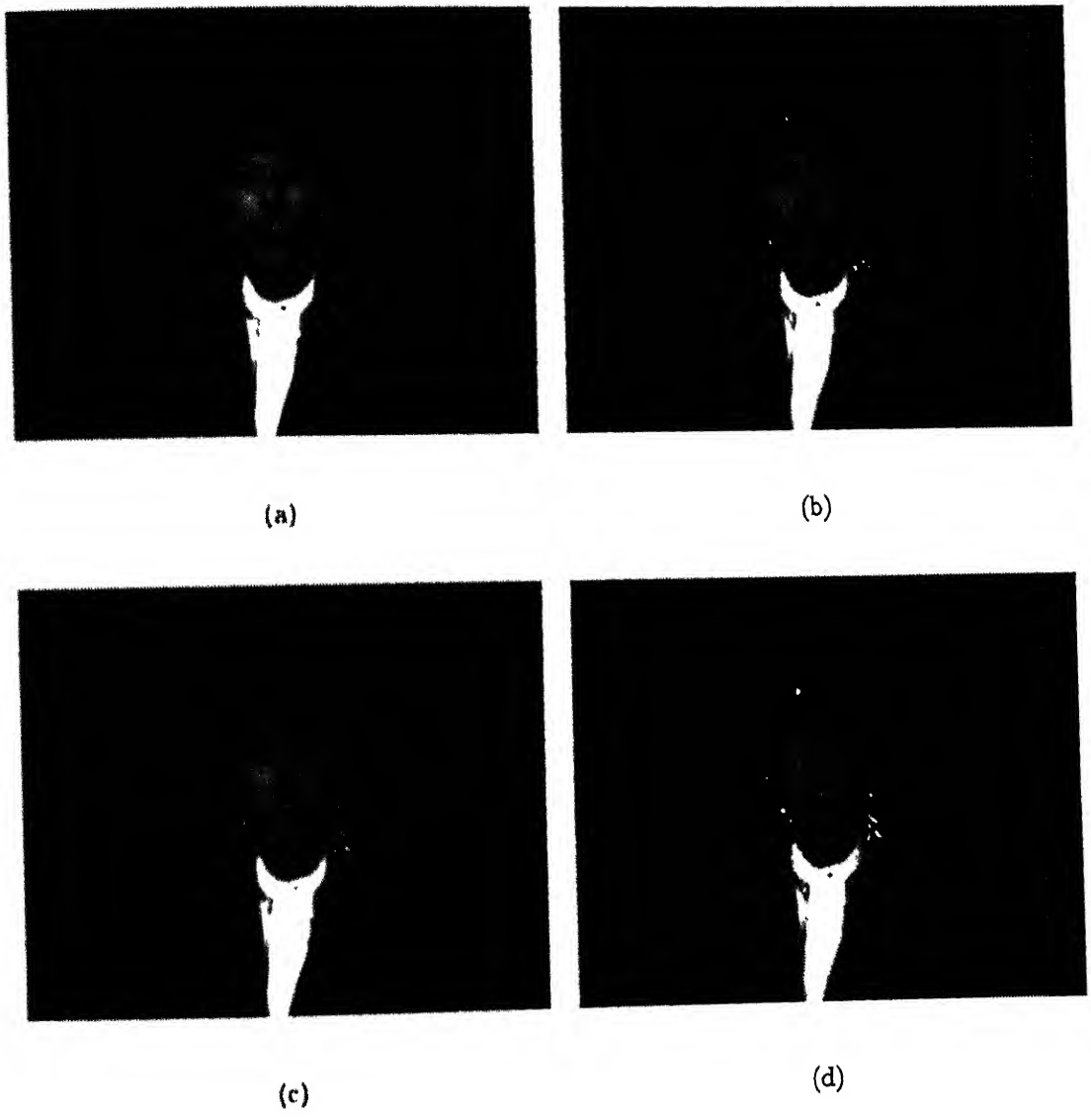


(a) (b)

(c) (d)

Figure 4.6: Seventh original frame (a),reconstructed frames (b) ,(c),(d) at b itrates 63kbps,61kbps,57kbps respectively

Comparison of Images at various bitrates



(a) (b)

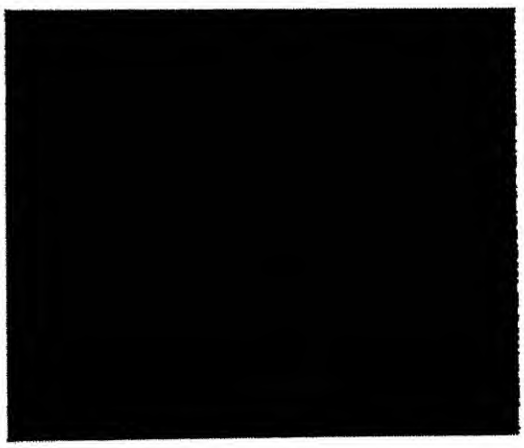
(c) (d)

Figure 4.7: Tenth original frame (a),reconstructed frames (b) ,(c),(d) at b itrates 63kbps,61kbps,57kbps respectively

(a) First original frame

(b) Reconstructed First frame at 68kbps

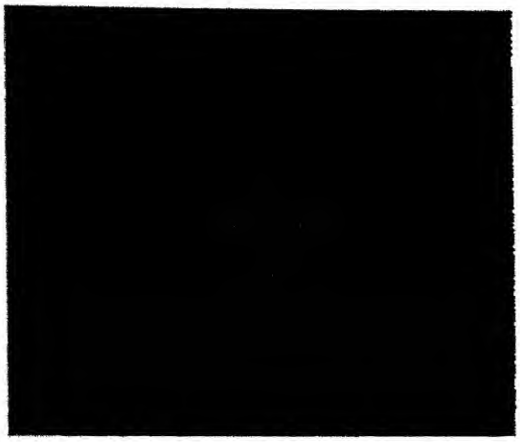
(c) Second original frame



(d) Reconstructed Second frame at 68kbps

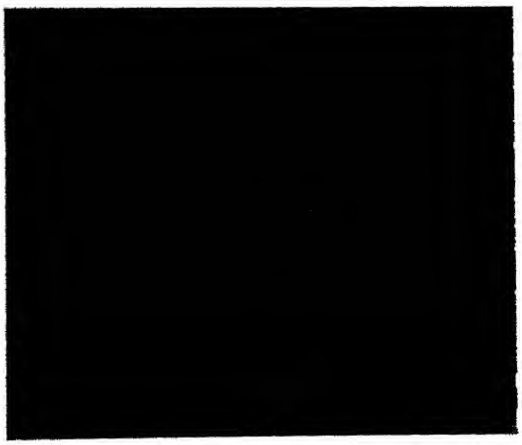
Figure 4.8: Comparison for Miss America Sequence

(a) Seventh original frame



(b) Reconstructed Seventh frame at 68kbps

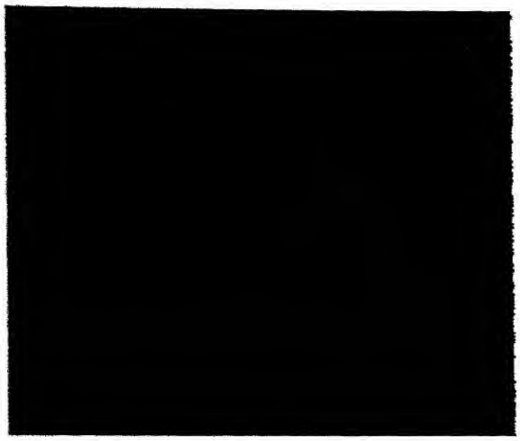
(c) Eighth original frame



(d) Reconstructed Eighth frame at 68kbps

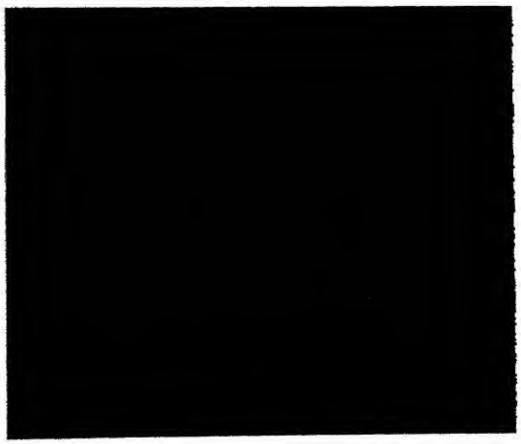
Figure 4.9: Comparison for Miss America Sequence

(a) The Second Frame of Salesman Sequence



(b) Reconstructed Second Frame at 81kbps

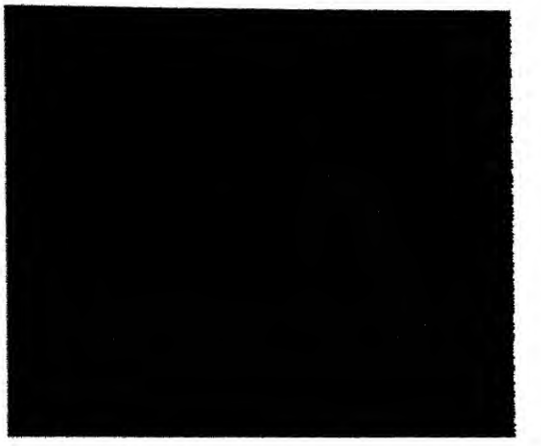
(c) The Third Frame of Salesman Sequence



(d) Reconstructed Third Frame at 81kbps

Figure 4.10: Original and Reconstructed Salesman Sequences

(a) The Sixth Frame of Salesman Sequence



(b) Reconstructed Sixth Frame at 81kbps

(c) The Seventh Frame of Salesman Sequence

(d) Reconstructed Seventh Frame at 81kbps

Figure 4.11: Original and Reconstructed Salesman Sequences

4.6 Discussions

In this section we consider two fidelity measures viz...,Model Failure area and PSNR to evaluate the reconstructed images.
Model Failure area is the percentage of pixels in the reconstructed frame where the mismatch is greater than 5.
PSNR is defined as

$$10 \log_{10} \frac{255^2}{MSE}$$

where MSE denotes the mean squared error of a pixel.

Referring to figures 4.12.4.13 and 4.14 we observe

- For a given frame the model failure area decreases as bitrates increase
- For a given frame the PSNR decreases as bitrate decreases
- The PSNR decreases as frame number increases
- Model Failure area increase as Frame number increases

Also we observe that bitrate was the lowest for claire sequence.The bitrate of the Salesman sequence was found to be higher than Miss America sequence The bitrates of the Miss America and Salesman sequences were fixed at 68kbps and 81kbps respectively,as further decrease of bitrates in these cases resulted in poor quality images.The values of deadzone and stepsize for the Miss America and Salesman sequences were 29 and 10 respectively.The quality of the reconstructed Claire sequence was found to be the best.The quality of the reconstructed Miss America sequence was better than the Salesman sequence.

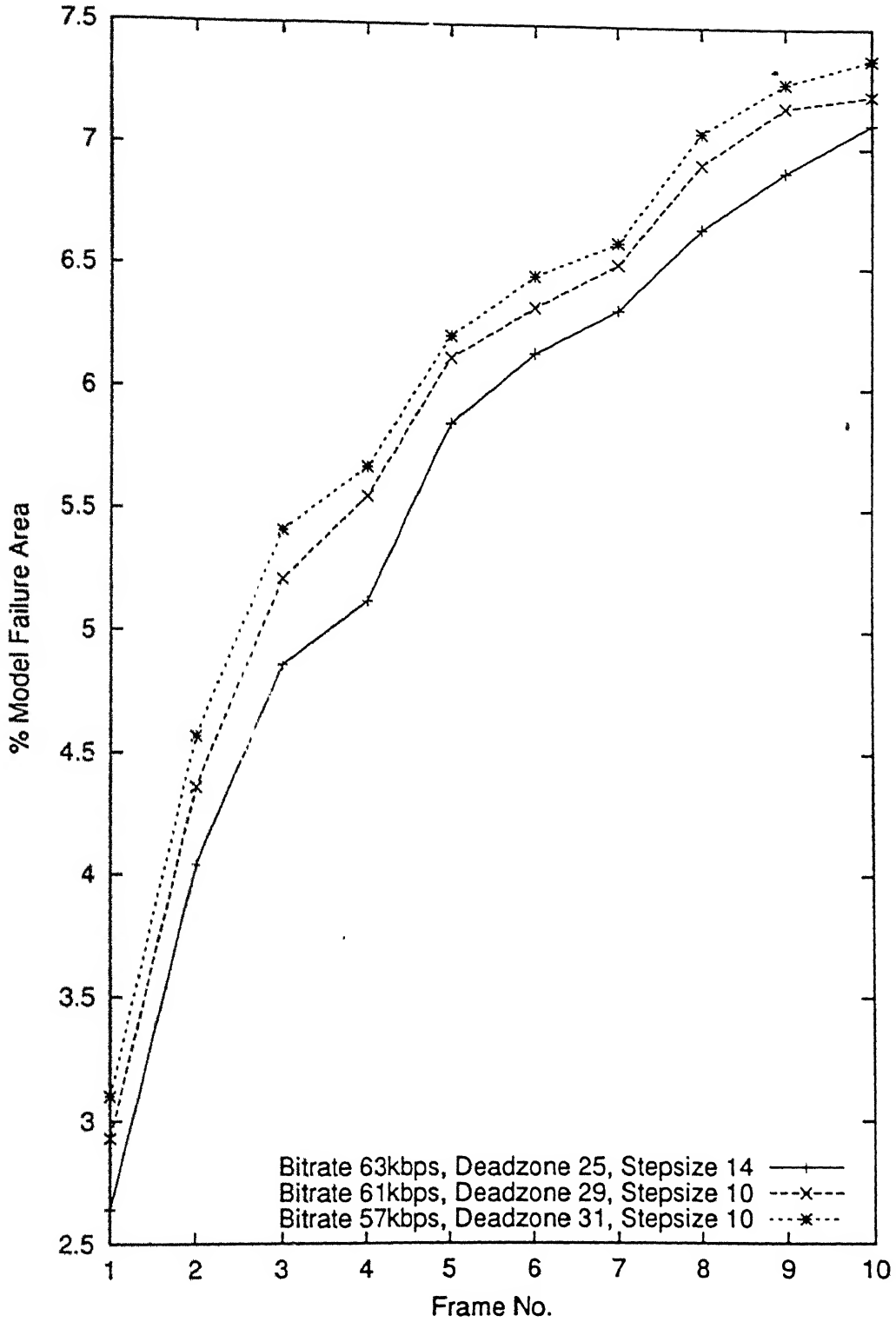


Figure 4.12: Percentage Model Failure Area v/s Frame number for Claire sequence at different bitrates

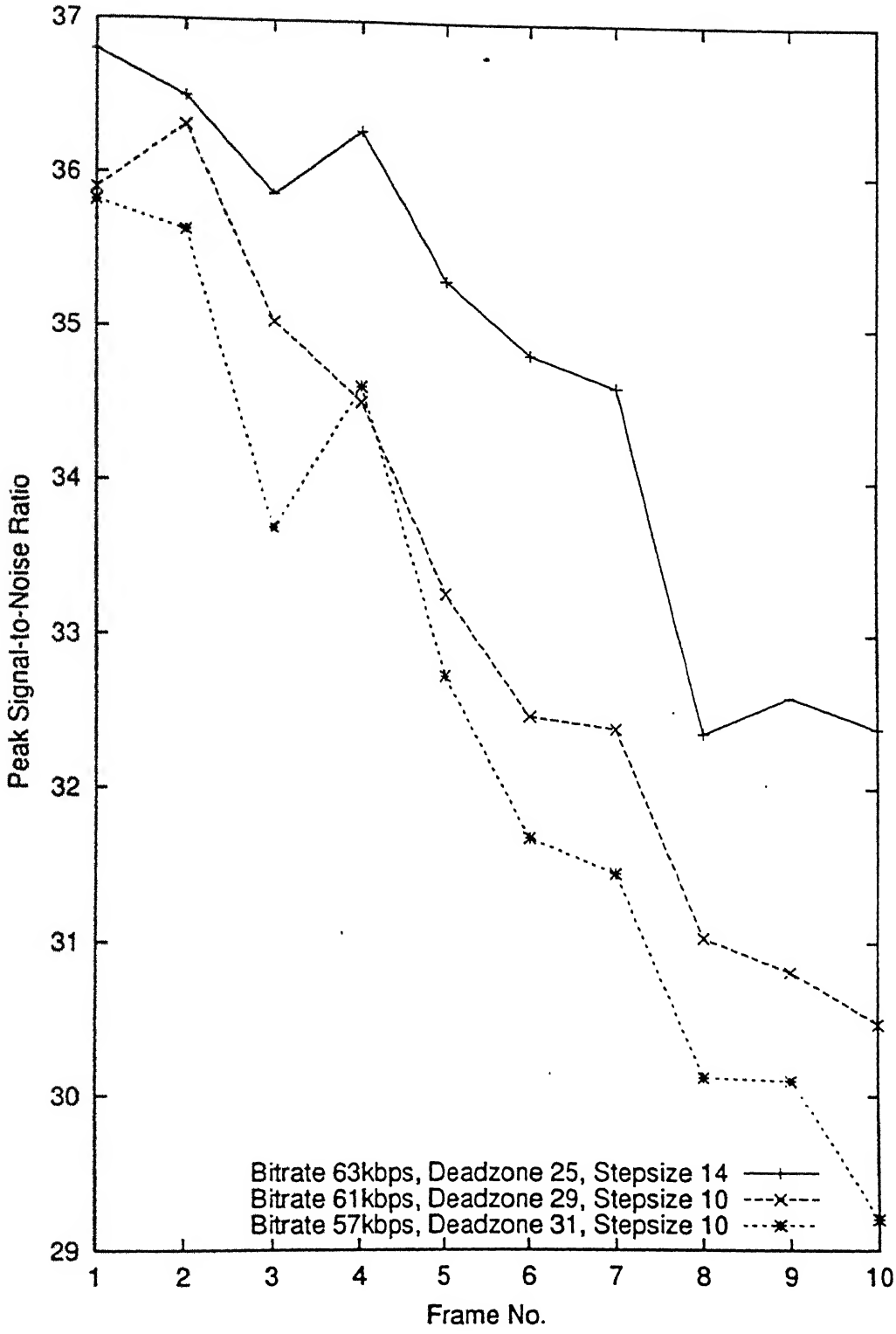


Figure 4.13: PSNR v/s Frame number for Claire Sequence at different bitrates

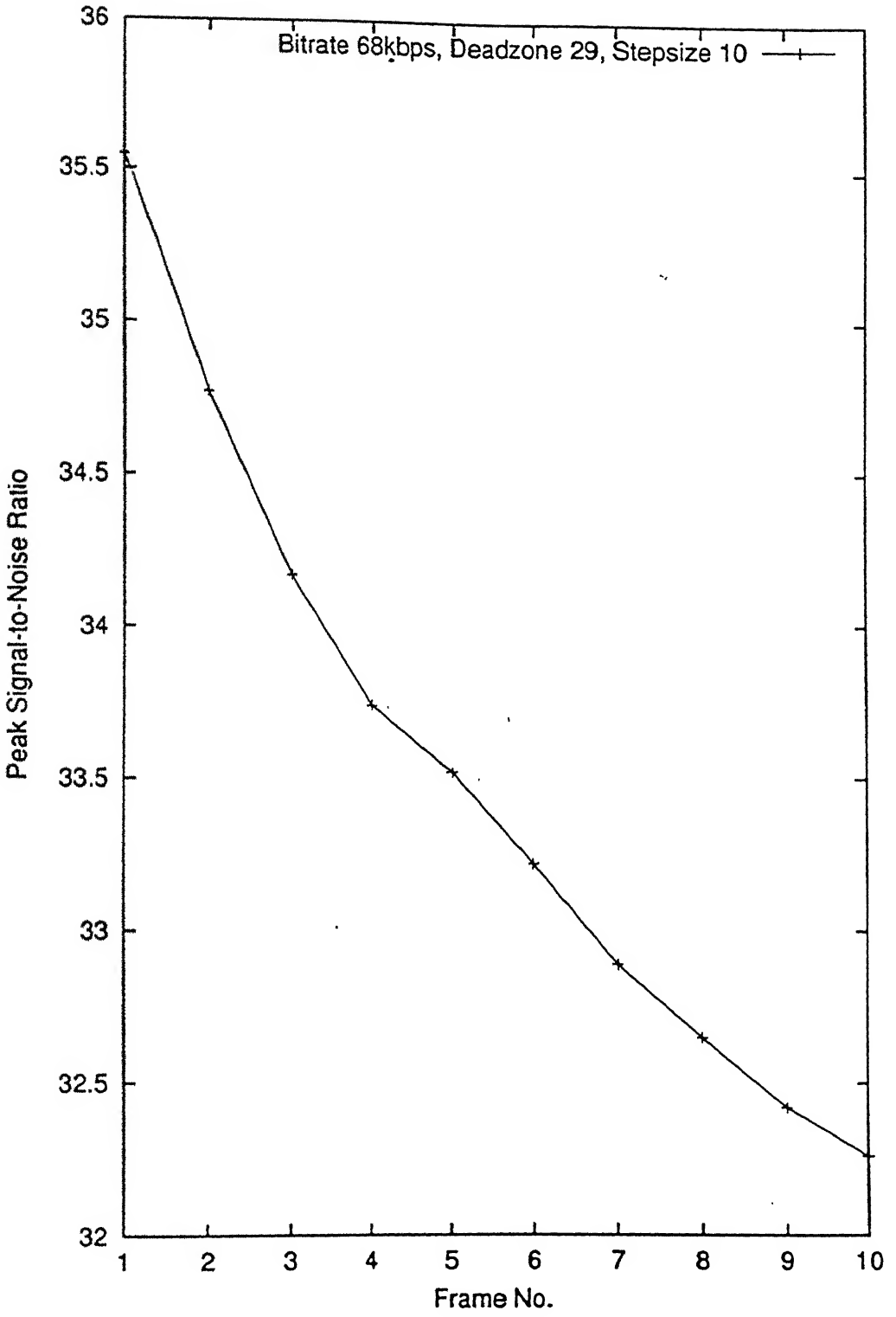


Figure 4.14: PSNR v/s Frame number for Miss America sequence

Chapter 5

Conclusions and Scope for future work

5.1 Conclusions

A low bitrate video coder was developed for a bitrate of 63kbps,61kbps,57kbps respectively. The method uses a new 3D motion estimation algorithm called the E Matrix method.Motion was estimated in a rectangular region enclosing the silhoutte.Once the motion was estimated in this region using the E Matrix method,the images were motion compensated.The next step was to generate the error of estimation.which is the difference between each image and it's estimated one.

This error was fed to the Video encoder which comprised of the SADCT, quantizer and VLC coder.The SADCT transforms a 8×8 block of pixels into transform coefficients.The SADCT had many advantages over the DCT.It was

found to be specially suited to code contours or the silhoutte.It also resulted in better quality of images at lower bitrates.The SADCT was also backward compatible with the standard DCT.

The quantizer converted the transform coefficients into one of the 255 levels ranging from -127 to 127.These levels for each 8×8 block were zigzag scanned and converted to serial bits using the VLC coder.The motion parameters were FLC coded.Both codes were sent as serial bits across the transmitting channel to the receiver.

On the receiving side these bits were decoded into levels and motion parameters by the VLD and FLD respectively.The levels were converted into transform coefficients by the Dequantizer.The SA-IDCT converted these transform coefficients into error values.From the motion parameters recovered by the FLD,the previous image in the Frame Store was motion compensated.The recovered error was added to this motion compensated image to get the final reconstructed image.This was passed through a Median Filter before display.

The video coder was tested on Claire,Miss America and Salesman sequences respectively.The PSNR ranges from 37 to 30 respectively while the Model Failure areas varied from 4 percent to 7 percent.

5.2 Scope for future work

The main aim of this thesis was to achieve low bitrates and good quality reconstructed images at the receiving side.There were many aspects which could have improved result.The main thing which influences the bitrate is motion es-

timation.It is found that motion estimation based on A Matrix method gives less error than the case where motion estimation is based on block matching.The present input to the E Matrix method is based on block matching.This should be replaced by the **A Matrix** method.

A lot of error analysis goes hand-in-hand with the E matrix method.Implementing this might reduce the bitrate.At present the motion compensation is done using the point correspondences obtained by the E Matrix method.The model,which is one of the outputs of the E Matrix method,is not used in motion compensation.Using the model to motion compensate the images might result in better quality.This can be done as follows.Get the 3D points before rotation from the E Matrix method.Using **Delaunay Triangularization** [1] construct a model of triangular patches from these 3D points.The coordinates of these triangles were got from the E Matrix method.Project the texture onto the model [1].The E matrix method also gives the 3D coordinates after motion.The motion of the 3D points can be looked upon as the motion of the model.Now back project the texture of the model onto the image plane.The resulting image is the motion compensated image.It remains to be seen whether the image so reconstructed is better than the present case or not.It is a general observation that all model based approaches give better quality and low bitrates than block matching based approaches.

The quantizer used is an uniform one.The compression is lossy because of the rounding off of the transform coefficients to levels in the quantizer.For a given number of levels it is known that the **Lloyd Max** quantizer gives the minimum error.In this quantizer the steps are non uniform and are designed to give minimum error due to rounding off of the transform coefficients into levels.Using such a quantizer would definitely improve the quality of images.

The deadzone and stepsizes should be dynamically varied so that they take smaller values in non-model failure regions.The bitrate may increase slightly but the quality of images would be better.This was shown in the preceding chapter.Other coding methods like Vectaor Quantization,Subband Coding etc can be explored to find the possibility of reducing the bitrate.

Chapter 6

Appendix

We present here the results for bit allocation .These results were carried out for two new bitrates.Also the reconstructed images are shown.The results shown is for the standard **Claire** sequence.The block size used in motion estimation is 19×19.

- Bitrate:41.5kbps
- Bits for Motion Parameters:2.5kbps
- Deadzone:35,stepsize:35
- PSNR(Frame Number):30.39(10),31.45(6)

- Bitrate:27.5kbps

- Bits for Motion Parameters:2.5kbps
- Deadzone:55,stepsize:35
- PSNR(Frame Number):29.78(10),28.97(5),
  29.78(4),32.19(2)

The reconstructed images at bitrates of 41.5kbps and 27.5kbps respectively are shown in Fig 6.1

(a) Tenth reconstructed image PSNR=30.39,Bitrate=41.5kbps

(b) Sixth reconstructed image PSNR=31.45,Bitrate=41.5kbps

(c) Tenth reconstructed image PSNR=29.78,Bitrate=27.5kbps

(d) Fifth reconstructed image PSNR=28.97,Bitrate=27.5kbps

Figure 6.1: Reconstructed images at bitrates of 41.5kbps and 27.5kbps respectively

Bibliography

[1] Narsi Reddy, "Design of low bitrate video coder", *IIT kanpur Mtech Thesis* 98/234MT,Dept of electrical engineering, 1998.

[2] Roger. Y. Tsai and Thomas. S .Huang, "Estimation of motion parameters of a rigid planar patch", *IEEE Trans on ASSP*, vol.29,No 6,p.1147-1152, Dec81.

[3] Roger. Y. Tsai and Thomas. S .Huang, "Estimating 3-Dimensional Motion Parameters of a Rigid Planar Patch,II" , *IEEE Trans on ASSP*, vol.30 No 4 August 82.

[4] Roger. Y. Tsai and Thomas. S .Huang, "Estimating 3-Dimensional Motion Parameters of a Rigid Planar Patch,III" , *IEEE Trans on ASSP*, vol.32 No 2 April84.

[5] Roger. Y. Tsai and Thomas. S .Huang, "Uniqueness and Estimation of 3-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces", *IEEE Trans on PAMI* vol 6 ,No 1,Jan 84.

[6] Thomas. S .Huang and Juang Weng, "Motion and Structure from 2 Perspective Views" , *IEEE Trans on PAMI* vol 2 ,No 5 ,May89.

[7] Jain A K, "Fundamenatals of Digital Image Processing", Prentice Hall of India, 1997.

[8] Gonzalez and Richard . F . Woods, " Digital Image Processing", Addison Wesley, 1993.

[9] Murat Tekalp, "Digital Video Processing", Prentice hall Signal Processing Series.

[10] Young T Y,King Sun Fu, "Handbook of Pattern Recognition and Image Processing" , Academic Press, 1989.

[11] Arun N Netravalli,Barry G Haskel, "Digital Pictures Representation,Compression and Standards", Plenum Press, 1995.

[12] Thomas Sikora, "MPEG Digital Video Coding Standards", *IEEE Signal Processing Magazine* Sep97.

[13] Thomas Sikora, "Shape Adaptive DCT for Generic Coding of Video", *IEEE Trans. on CSVT* , vol.5, no.1, Feb95.

[14] Hamiton W R, "Elements of Quaternions", 3^{rd} edition New York Chelsea 1969.

[15] W. Press, "Numerical Recipes in C," Cambridge University Press, 1992.

[16] Hwang J J,Rao K R, "Techniques and Standards for Image,Video and Audio Coding", Prentice Hall, New Jersey 1996 Edition.

[17] Botteme O, Roth B "Theoretical Kinematics", New York: North Holland, 1979.